

# ***SpaceSched: A Constellation-Wide Scheduling System for Resolving Ground Track Congestion in Remote Sensing***

Zehua Sun<sup>†</sup>, Tao Ni<sup>†</sup>, Pengfei Hu<sup>‡</sup>, Tao Gu<sup>§</sup>, Weitao Xu<sup>†\*</sup>

<sup>†</sup>City University of Hong Kong, Hong Kong, China

<sup>‡</sup>Shandong University, Qingdao, China <sup>§</sup>Macquarie University, Sydney, Australia

Email: zehua.sun@my.cityu.edu.hk, phu@sdu.edu.cn, tao.gu@mq.edu.au, {taoni2, weitaoxu}@cityu.edu.hk

## **ABSTRACT**

The recent proliferation of spacecraft in Earth’s orbits has ushered in the rise of large-scale satellite constellations. However, this unprecedented growth of constellations has introduced a previously unforeseen challenge: ground track congestion. Specifically, the increasing density of orbital slots forces satellites to share similar orbit planes, causing their nadir-point projections on Earth’s surface (i.e., ground tracks) to overlap or remain in close proximity within short time intervals. Such orbit-endowed ground track congestion can degrade constellation performance in remote sensing operations, specified by limited constellation coverage, redundant satellite count, and delayed data delivery.

To address this issue, we propose *SpaceSched*, a hierarchical scheduling framework designed to resolve ground track congestion in satellite constellations. *SpaceSched* consists of two key components: an on-ground constellation scheduling pipeline, comprising a coverage distributor and a satellite selector, and an in-space downlink scheduling pipeline, featuring a queue regulator. The coverage distributor assigns attitude profiles over time to satellites, ensuring non-overlapping imagery capture regions. The satellite selector optimizes the constellation by strategically selecting a subset of satellites while maintaining coverage efficiency. During in-space downlink scheduling, the queue regulator manages the downlink traffic queue to minimize the delay of high-value data transmission. We evaluate *SpaceSched*

\* Weitao Xu is the corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ACM MOBICOM ’25, November 4–8, 2025, Hong Kong, China  
© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 979-8-4007-1129-9/25/11...\$15.00  
<https://doi.org/10.1145/3680207.3765249>

on two operational modes (i.e., stripmap and spotlight) across three well-established satellite constellation systems: SKYSAT, LEMUR, and FLOCK, with 17, 50, and 126 evaluated satellites, respectively. Experimental results demonstrate that *SpaceSched* improves coverage by up to 1.84×, reduces satellite counts by up to 2.38×, and decreases downlink queue load by up to 36.46×, compared to the plain satellite constellation systems. Furthermore, our case study highlights *SpaceSched*’s potential to meet diverse task demands.

## **CCS CONCEPTS**

• Networks; • Computing methodologies;

## **KEYWORDS**

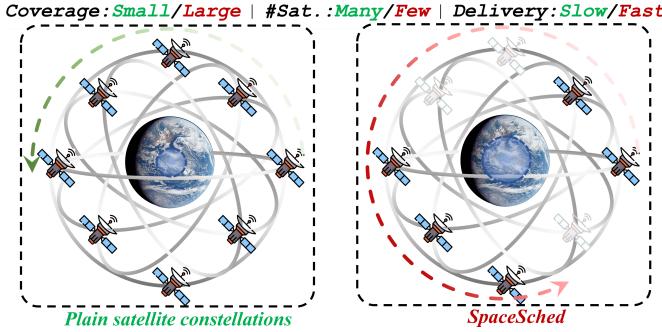
Satellite Constellation, Remote Sensing, Scheduling

## **ACM Reference Format:**

Zehua Sun, Tao Ni, Pengfei Hu, Tao Gu, Weitao Xu. 2025. *SpaceSched: A Constellation-Wide Scheduling System for Resolving Ground Track Congestion in Remote Sensing*. In *The 31st Annual International Conference on Mobile Computing and Networking (ACM MOBICOM ’25), November 4–8, 2025, Hong Kong, China*. ACM, New York, NY, USA, 16 pages. <https://doi.org/10.1145/3680207.3765249>

## **1 INTRODUCTION**

Over the past few decades, the rapid expansion of space-craft deployments has driven the rise of large-scale satellite constellations across various applications, including remote sensing (e.g., Landsat [76], Sentinel [59]), communication (e.g., Starlink [51], OneWeb [29]), and navigation (e.g., GPS [47], GLONASS [54]). However, this rapid proliferation has intensified a growing challenge: orbital congestion [3, 5, 11, 57], particularly in Low Earth Orbits (LEOs). As highlighted by Slingshot Aerospace, “Congestion in Earth’s orbit is getting even worse” [1]. By the end of 2023, more than 12,597 satellites were in Earth orbit, with over 91% operating in LEOs. Alarmingly, more than 3,300 satellites remain idle, occupying valuable orbital slots and exacerbating congestion. Addressing this issue directly is challenging, as



**Figure 1: Illustration of SpaceSched.** During on-ground constellation scheduling, the coverage distributor improves the constellation coverage ( $1.84\times$ ), and the satellite selector further reduces the satellite count ( $2.38\times$ ). During in-space downlink scheduling, the queue regulator decreases the data delivery load of downlink traffic ( $36.46\times$ ).

it requires either physically removing space debris or relocating satellites to higher-altitude orbits [35, 73].

**Ground track congestion.** Such orbital congestion challenges compel deployed satellites to share more comparable or adjacent orbital planes (i.e., similar orbit elements) for orbit slot saving [4, 79]. A striking example of this issue occurred in 2009 when an Iridium 33 and a Russian military Cosmos satellite collided—an accident largely attributed to orbital overcrowding in near-polar orbits at 789 km altitude and  $86^\circ$  inclination, which are heavily utilized by commercial constellations [21]. The scarcity of premium orbital slots exacerbates this problem, particularly in  $98^\circ$  sun-synchronous orbits (e.g., America Landsat, France SPOT, and Canada RADARSAT), due to their ability to provide consistent illumination and precise revisit cycles. Consequently, the nadir-point projections of these satellites on Earth’s surface (i.e., ground tracks) overlap or remain in close proximity within short time intervals (see Figure 5). We introduce and define this phenomenon as **ground track congestion**. Unlike orbital congestion [5, 57], where satellites overcrowd within orbital slots, ground track congestion is more prominent due to the altitude-agnostic nature of satellite projections and the high dynamics of orbital movement. This congestion significantly degrades satellite constellation performance in remote sensing operations [83], manifesting in three key aspects: **1) Limited constellation coverage** – Overlapping imagery regions reduce the overall efficiency of data collection. **2) Redundant satellite count** – A lack of priority differentiation leads to unnecessary satellites operating in the same coverage area. **3) Delayed data delivery** – Narrow downlink bandwidth forces high-value data to compete for transmission, leading to delivery delays [15, 80]. Ultimately, ground track congestion wastes space resources, increases operational costs, and hampers the constellation’s service to deliver timely and diverse data, making it a critical implication for satellite operators.

**Prior research and comparison.** As illustrated in Table 1, the existing solutions for mitigating congestion can be categorized into organization methods and scheduling methods. Orbit-level organization methods [37, 52] focus on optimizing sophisticated orbit slot planning and orbit element designs to reduce spatial conflicts targeting satellites yet to be launched. While these methods could directly address ground track congestion, they are costly and constrained by limited space resources and regulatory complexities. Other methods typically prioritize specific performance metrics rather than directly tackling ground track congestion. For example, ground station distribution organization methods [20, 71] primarily enhance the downlink traffic. Meanwhile, scheduling methods rely on specialized hardware components (e.g., Reconfigurable Intelligent Surface (RIS) [26], antenna array [85]) or satellite software-level configurations to optimize metrics in terms of coverage [39, 68], transmission [38, 70], and Quality of Service (QoS) [30, 44]. Moreover, these solutions often overlook critical constraints such as constellation resources, diverse task demands, and power limitations. Consequently, ground track congestion remains an inadequately addressed challenge, which underscores a significant research gap.

As the first trial, we propose *SpaceSched*, a hierarchical scheduling framework designed to resolve ground track congestion in satellite constellations from three key perspectives. As illustrated in Figure 1, *SpaceSched* comprises two main scheduling pipelines: an on-ground constellation scheduling pipeline, which includes a coverage distributor and a satellite selector, and an in-space downlink scheduling pipeline, featuring a queue regulator. On the ground, the coverage distributor dynamically assigns attitude profiles to each satellite, reorienting them to distribute imagery capture tracks on non-overlapping regions. The satellite selector further refines the constellation by elaborately selecting a subset of satellites without compromising the attained coverage performance. In space, the queue regulator manages the downlink traffic queues of the selected satellites, prioritizing high-value data to minimize transmission load and delay. Notably, *SpaceSched* operates entirely at the software level, leveraging well-established attitude control and in-orbit edge computing capabilities for seamless deployment (see § 2.1).

To design and implement *SpaceSched*, we have addressed the following three key challenges:

**Challenge 1: Constellation-wide coordination.** Multiple satellites within the constellation exhibit variations in motion dynamics (i.e., spatial-temporal coordinates) and task status (i.e., attitude-induced spatial coverage). These variations present the space-time-attitude triad coordination challenges. 1) Specifically, satellites within

**Table 1: Comparison with the existing works.** CR: constellation resource, SR: satellite resource, ED: environmental dynamics.

	Method	CR	SR	Load	Link	Task	Power	ED
Organization	Orbit-level	Maqrot et al. [52]	○	●	○	○	○	○
		Laguna et al. [37]	●	●	○	○	○	●
Scheduling	GS-level	Efrem et al. [20]	○	●	●	○	○	○
		DGS [71]	○	●	●	●	○	●
Hardware-level	SpaceRIS [26]	●	●	○	●	○	●	●
	Zhu et al. [85]	○	●	○	●	○	●	○
	Umbra [70]	○	●	●	●	●	○	●
	Li et al. [44]	●	●	●	●	●	○	●
Software-level	STECN [30]	○	●	●	●	●	●	○
	Lei et al. [38]	○	●	○	●	○	●	○
	<i>SpaceSched</i> (Ours)	●	●	●	●	●	●	●

(●—satisfactory, ▲—moderate, ○—weak)

the constellation have their respective orbital states issued at different timestamps, leading to spatial-temporal discrepancies across the constellation. To address this, *SpaceSched* applies the nearest-time principle to establish a unified temporal reference for time synchronization. It then compensates for time offsets using the “epoch day” format [34], ensuring consistent spatial alignment within the constellation. 2) Furthermore, satellites in orbit do not maintain a fixed attitude configuration, causing fluctuations in spatial coverage. *SpaceSched* defines the spatial scope via boundary offsets and center shifts, then performs attitude-aware calibration to determine the final coverage.

**Challenge 2: Hierarchical scheduling discrepancy.** During on-ground constellation scheduling, satellites within a constellation typically have different priorities based on their contributions to constellation coverage and satellite count. 1) Constellation coverage: Satellites are strategically assigned to cover non-overlapping regions to maximize efficiency. However, beyond their fixed ground tracks, further challenges arise, particularly meteorological dynamics and attitude variation intensity. Meteorological dynamics [45, 82] can significantly degrade data quality, necessitating prioritization of contamination-free regions. Moreover, frequent and abrupt changes [32] consume substantial energy and accelerate mechanical wear, making it crucial to maintain stable attitude variations. To address these challenges, the coverage distributor formulates a multi-objective optimization model to incorporate these factors as constraints and penalties. Furthermore, since not all satellites in the constellation are fully engaged at all times, optimizing coverage with partial satellite availability becomes an additional challenge. To overcome this, *SpaceSched* applies the increment-accumulation principle, independently assessing and aggregating each satellite’s coverage contributions. 2) Satellite count: Identifying the optimal subset of active satellites that maximally contribute to coverage is another challenge. Determining individual satellite priorities and assessing the impact of coverage sacrifices is complex. To tackle this, the satellite selector formulates

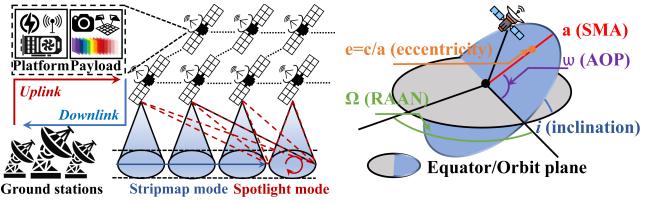


Figure 2: Satellite constellations.

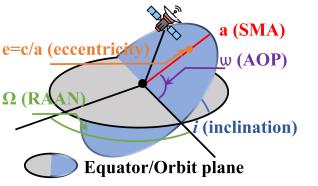


Figure 3: Orbit elements.

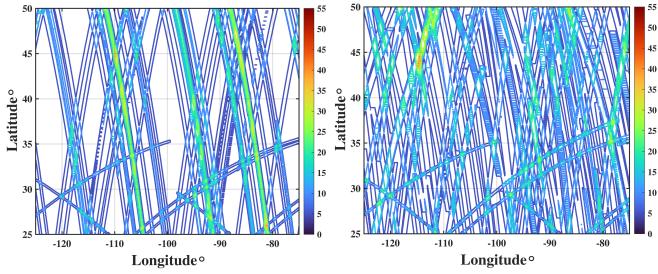
a coverage-preserving model, incorporating coverage loss penalties as hard constraints, and employs a combinatorial search approach to determine the optimal satellite subset.

**Challenge 3: Downlink traffic imbalance.** Satellites generate vast amounts of data, but their downlink traffic is often imbalanced at both the intra-satellite (i.e., high-value data ratio over time) and inter-satellite levels (i.e., distribution across satellites). Furthermore, satellites face significant constraints in downlink bandwidth bottleneck, further exacerbated by the high orbital dynamic nature and the sparse distribution of ground stations, rendering only 2% of the data received [15, 16]. To address this challenge, the queue regulator prioritizes the transmission of high-value data at the intra-satellite level, which leverages the global coverage map generated by the satellite selector as prior knowledge. Furthermore, to ensure load fairness across satellites, a fairness-aware combinatorial optimization strategy is applied at the inter-satellite level.

We implement *SpaceSched* using real-world Two-Line Element (TLE) orbit descriptors, and evaluate its performance on two modes (i.e., stripmap and spotlight) across three well-established satellite constellation systems: SKYSAT, LEMUR, and FLOCK, consisting of 17, 50, and 126 evaluated satellites, respectively. Experimental results demonstrate that *SpaceSched* improves coverage by up to 1.84×, reduces satellite counts by up to 2.38×, and decreases downlink queue load by up to 36.46×, compared to conventional satellite constellation systems. Furthermore, our case study highlights *SpaceSched*’s potential to meet diverse task demands.

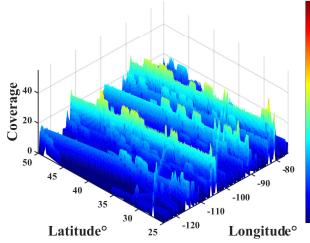
Overall, *SpaceSched* makes the following contributions:

- We define the phenomenon of ground track congestion, which results in suboptimal constellation performance, including limited constellation coverage, redundant satellite count, and delayed data delivery.
- We propose *SpaceSched*, a hierarchical scheduling framework to resolve ground track congestion. *SpaceSched* consists of two key components: an on-ground constellation scheduling pipeline, comprising a coverage distributor and a satellite selector, and an in-space downlink scheduling pipeline, featuring a queue regulator.
- The extensive experimental results demonstrate that *SpaceSched* improves coverage by up to 1.84×, reduces satellite counts by up to 2.38×, and decreases downlink queue load by up to 36.46×.



(a) The plain satellite constellation.

(b) SpaceSched.



(c) The plain satellite constellation.

(d) SpaceSched.

Figure 4: Illustration of ground track congestion (top: 2D plane, bottom: 3D plane). The plain constellation exhibits limited and uneven coverage, while *SpaceSched* mitigates ground track congestion.

## 2 PRELIMINARY & MOTIVATION

### 2.1 Satellite Constellation Primer

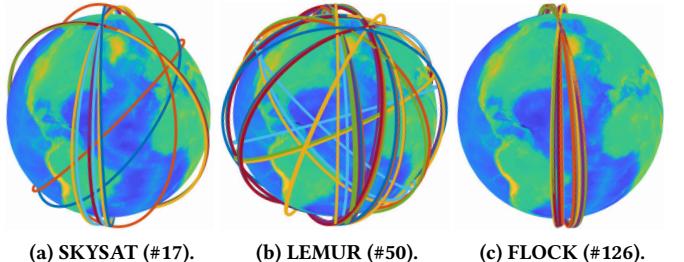
**Satellite constellation.** As shown in Figure 2, a remote sensing satellite constellation is a strategically coordinated group of satellites deployed in specific orbital patterns to collectively perform tasks. Satellites collect data in orbit and transmit it to ground stations via downlink, while ground stations use uplink to send control commands.

**Satellite system.** Each satellite typically consists of two primary components: the platform and the payload. The platform provides the core systems necessary for operation, such as power supply, attitude control, and computing hardware. Among these, the Attitude and Orbit Control System (AOCS) allows precise three-axis stabilized attitude control [36, 68]. Furthermore, the computing hardware provides satellites with edge computing capabilities in orbit for in-space downlink scheduling. Meanwhile, the payload carries specialized equipment designed to fulfill various task demands, such as optical cameras, multispectral sensors, and Synthetic Aperture Radars (SAR).

**Mode.** The primary imagery modes include stripmap and spotlight. Stripmap mode, the standard operational mode, captures a continuous swath along the satellite’s ground track. In contrast, spotlight mode steers the sensor to remain on a specific target area for a relatively long period.

**Orbital element.** As shown in Figure 3, the orbits (e.g., size, shape, orientation, and position) are determined by the six orbital element parameters [63]:

- The semi-major axis  $a$  determines the orbit’s size.
- The eccentricity  $e$  indicates the orbit’s flatness.



(a) SKYSAT (#17). (b) LEMUR (#50). (c) FLOCK (#126).

Figure 5: Illustration of orbital congestion.

- The inclination  $i$  describes the orbit’s tilt relative.
- The argument of perigee  $\omega$  is the angle from the ascending node to the periapsis in the orbital plane.
- The right ascension of the ascending node  $\Omega$  is the angle from the vernal equinox to the ascending node in the reference plane.
- The mean anomaly  $M$  indicates the orbital period fraction since the last periapsis passage.

### 2.2 Ground Track Congestion

To investigate ground track congestion, we conduct a quantitative comparison of constellation coverage between a plain satellite constellation and *SpaceSched* in Figure 4. Specifically, Figure 4a & Figure 4c reveal that the coverage is primarily concentrated along the ground tracks. However, the ground tracks of these satellites are unevenly distributed: some regions experience excessive imagery, while others suffer under-observed or entirely unmonitored. Moreover, real-world imagery acquired by Planet Labs further confirms this effect [22], serving as cross-validation of the ground track congestion. In comparison, *SpaceSched* mitigates ground track congestion, improving coverage from 55.17% to 84.24%, as evidenced in Figure 4b & Figure 4d. This phenomenon primarily stems from the satellite oversaturation distribution in orbit slots, as shown in Figure 5, which presents significant challenges in limited constellation coverage, redundant satellite count, and delayed delivery of high-value data.

## 3 SYSTEM OVERVIEW

Figure 6 presents an overview of *SpaceSched*, which operates across both ground stations and satellite constellations.

**Ground stations.** Upon issuing orbital descriptors of satellite constellations, *SpaceSched* initiates constellation-wide coordination (§ 4), i.e., space-time-attitude triad coordination. Specifically, *SpaceSched* leverages a unified temporal reference for time synchronization, and derives the time offset for space compensation. *SpaceSched* then calibrates the satellite spatial coverage based on attitude-induced boundary offset and center shift. After constellation-wide coordination, *SpaceSched* relies on an on-ground constellation scheduling pipeline (§ 5) consisting of a coverage distributor

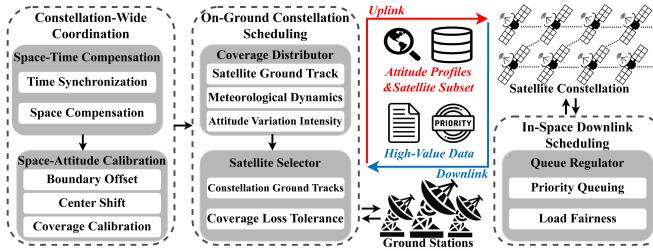


Figure 6: Overview of SpaceSched (§ 3).

(§ 5.1) and a satellite selector (§ 5.2). The coverage distributor optimizes the constellation coverage profit by factoring in satellite ground track, meteorological conditions, and attitude variation intensity. Through increment-accumulation scheduling, the coverage distributor assigns attitude profiles (i.e., yaw, pitch, and roll angles) over time to each satellite within the constellation. Furthermore, the satellite selector further refines the constellation count through combinatorial search without compromising the attained coverage performance. This process is executed periodically to guarantee that the constellation remains in a continuously optimized operational state.

**Satellite constellations.** SpaceSched incorporates an in-space downlink scheduling pipeline (§ 6) featuring a queue regulator (§ 6.1). The active satellites, upon selection, are configured with assigned attitude profiles and a global coverage map as prior knowledge. The queue regulator prioritizes the transmission of high-value data at the intra-satellite level with this map as the global perspective, and further ensures equitable load (§ 6.2) at the inter-satellite level through fairness-aware combinatorial optimization.

## 4 CONSTELLATION-WIDE COORD.

A constellation comprises multiple satellites, each exhibiting variations in motion dynamics (i.e., spatial-temporal coordinates) and task status (i.e., attitude-induced spatial coverage). Thus, ground stations require constellation-wide coordination to eliminate such inconsistencies before executing ground-space scheduling, i.e., space-time compensation (§ 4.1) and space-attitude calibration (§ 4.2).

### 4.1 Space-Time Compensation

SpaceSched requires the latest orbital state of each satellite within the constellation for precise scheduling. However, satellites typically have their respective orbital states issued at different timestamps by ground stations, which leads to severe time desynchronization [42, 74] and spatial offsets across the constellation. To address this problem, SpaceSched performs space-time compensation, manifested by time synchronization and space compensation.

**Time synchronization.** Each satellite's orbital state is represented by the TLE descriptor [34] published by

ground-based tracking catalogs, which records the orbital elements specifying the satellite's position and motion characteristics. These orbital states are timestamped under the “epoch day” format, representing the day of the year and the fractional part of the day as a decimal value.

Formally, we represent a satellite constellation with  $n$  satellites as  $SAT = \{S_1, S_2, \dots, S_n\}$ , with their recording timestamp “epoch day” as  $TLE = \{TLE_1, TLE_2, \dots, TLE_n\}$ . To establish a unified temporal reference, we adopt the nearest-time synchronization principle, where the closest “epoch day” to the present time is selected as the constellation’s reference time  $TLE_r$ . This synchronization process yields a time offset vector  $\Delta T = TLE_r - \{TLE_1, TLE_2, \dots, TLE_n\}$  for the constellation.

**Space compensation.** We define time  $t \in T$  as the time point during the entire scheduling process. To compute the spatial location of a satellite at time  $t$  with time offset  $\Delta t$ , we figure mean anomaly  $M$ , eccentric anomaly  $E$ , and true anomaly  $V$  out in turn [31]. Specifically, the mean anomaly  $M$  is represented as:

$$M(t) = M_0 + n(t + \Delta t - t_0), \quad (1)$$

where  $M_0$  is the mean anomaly at the reference epoch  $t_0$ , and  $n$  represents mean angular motion, calculated as  $n = \sqrt{\mu/a^3}$ . With  $M$ , we solve Kepler’s Equation to relate the mean anomaly  $M$  to the eccentric anomaly  $E$ :

$$M(t) = E(t) - e \sin E(t). \quad (2)$$

The true anomaly  $V$ , which gives the actual angular position in its orbit, is then expressed as:

$$\tan(V(t)/2) = \sqrt{(1+e)/(1-e)} \tan(E(t)/2). \quad (3)$$

With the true anomaly  $V$ , we can determine the satellite’s position in the orbital plane:

$$\{ x' = r \cos(V), y' = r \sin(V), z' = r \cdot 0, \quad (4)$$

where the radius  $r$  represents the distance between the focus of attraction and the satellite, expressed as  $r = a(1-e^2)/(1+e \cos(V))$ . The compensated coordinates are then transformed into the Earth’s longitude, latitude, and altitude coordinates  $(\lambda(t), \phi(t), h(t))$  through coordinate transformation [62].

### 4.2 Space-Attitude Calibration

Satellites in orbit do not maintain a fixed attitude configuration, which is actively controlled by AOCS to accommodate diverse task objectives and operational requirements. However, when the satellite attitude varies, i.e., yaw, pitch, and roll angles, it can significantly alter the Field of View (FoV), resulting in substantial spatial coverage fluctuations. To address such attitude-induced coverage deviations, SpaceSched calibrates the coverage by scoping FoV boundary offset and center shift.

**Attitude.** Attitude is typically described in three-dimensional Euler angles, i.e., yaw ( $\psi$ ), pitch ( $\theta$ ), and roll ( $\xi$ ). These angles denote rotations about the vertical (Z), lateral (Y), and longitudinal (X) body axes, respectively. The attitude matrix,  $\mathbf{A}$ , describes the rotation from the satellite's body frame to the Earth-Centered Inertial (ECI) frame. This matrix is computed as the product of three fundamental rotation matrices, represented as:

$$\mathbf{A} = \mathbf{R}_z(\psi)\mathbf{R}_y(\theta)\mathbf{R}_x(\xi), \quad (5)$$

$$\mathbf{R}_z(\psi) = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix}, \mathbf{R}_y(\theta) = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix}, \mathbf{R}_x(\xi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \xi & -\sin \xi \\ 0 & \sin \xi & \cos \xi \end{bmatrix}.$$

**FoV.** The FoV refers to the angular extent of the region that the onboard instrument can capture at a given moment. It is dictated by the sensor's optics configuration as  $\text{FoV} = 2 \cdot \arctan(d/2f)$  [33], where  $d$  is the sensor's dimension (width or height) and  $f$  is the focal length. Thus, the coverage on the Earth's surface is expressed as:

$$R_h = h \cdot \tan\left(\frac{\text{FoV}_h}{2}\right), \quad R_v = h \cdot \tan\left(\frac{\text{FoV}_v}{2}\right), \quad (6)$$

where  $h$  is the altitude, and  $\text{FoV}_h$  and  $\text{FoV}_v$  are the horizontal and vertical FoVs, respectively.

**Boundary offset.** In this paper, we define attitude as the payload boresight w.r.t. local nadir, where  $\psi, \theta$  denote cross-/along-track look angles in the local ENU frame. Specifically, the yaw angle  $\psi(t)$  governs the satellite's left-right rotation, directly influencing the horizontal projection of FoVs. The pitch angle  $\theta(t)$  controls the satellite's upward-downward rotation, impacting the vertical projection of FoVs. Consequently, the adjustments to longitude and latitude projections, factoring in the yaw/pitch angles, can be approximated as:

$$R'_h = \frac{R_h}{\cos(\psi(t))}, \quad R'_v = \frac{R_v}{\cos(\theta(t))}, \quad (7)$$

where  $R'_h$  and  $R'_v$  represent the adjusted horizontal and vertical coverage region ranges, respectively. The roll angle  $\xi(t)$  describes the satellite's rotation around its longitudinal axis. Since this rotation does not alter the coverage of a single shot, its influence is not considered in this paper.

**Center shift.** Apart from the boundary offset, the FoV center shifts accordingly. Given the original FoV center at the satellite's nadir point,  $(\lambda(t), \phi(t), h(t))$ , the shifted center can be calculated as follows:

$$\lambda(t)' = \lambda(t) + \frac{h(t) \cdot \tan(\psi(t))}{R_e \cdot \cos(\phi(t))}, \quad \phi(t)' = \phi(t) + \frac{h(t) \cdot \tan(\theta(t))}{R_e}, \quad (8)$$

where  $(\lambda', \phi')$  represents the shifted FoV center, and  $R_e$  is the radius of the Earth.

**Coverage calibration.** Incorporating the boundary offset and center shift, the longitude  $\Delta\lambda$  and latitude half-extents  $\Delta\phi$  scoping the final coverage are given by

$$\Delta\lambda = \frac{R'_h}{R_e \cdot \cos(\phi'(t))}, \quad \Delta\phi = \frac{R'_v}{R_e}. \quad (9)$$

Accordingly, the final coverage (i.e., imagery region) is  $([\lambda' - \Delta\lambda, \lambda' + \Delta\lambda], [\phi' - \Delta\phi, \phi' + \Delta\phi])$ .

## 5 ON-GROUND CONSTELLATION SCHED.

### 5.1 Coverage Distributor

The first issue arising from ground track congestion is the sub-optimal coverage of satellite constellations. However, distributing the imagery capture tracks across non-overlapping regions presents significant challenges, particularly meteorological dynamics and attitude variation intensity. Although task-driven coverage expansion works have been extensively investigated [26, 39, 68, 85], they neglect partial satellite availability, i.e., not all satellites within the constellation are operationally engaged. To overcome these limitations, *SpaceSched* designs a coverage distributor, which well handles trade-offs of optimizing coverage while minimizing contamination region observations and attitude variation intensity (§ 5.1.1), through scheduling in an increment-accumulation manner (§ 5.1.2).

**5.1.1 MOOP Formulation.** In the coverage distributor, the decisions are based on the following factors:

- **Satellite ground track.** Each satellite within the constellation follows a distinct ground track and possesses its respective coverage regions. Thus, the primary goal is to maximize constellation coverage by minimizing overlapping ground tracks across satellites.
- **Meteorological dynamics.** The data quality of satellite imagery can be severely degraded by meteorological dynamics, such as cloud cover/shadow and other atmospheric conditions. Thus, *SpaceSched* is required to prioritize capturing contamination-free regions.
- **Attitude variation intensity.** Frequent or abrupt changes in satellite attitude consume substantial amounts of energy and accelerate wear on mechanical components, particularly for space IoT devices with limited power resources and high maintenance costs. Therefore, a well-designed attitude control strategy balances the satellite coverage and energy efficiency.

**Satellite ground track.** Each satellite follows a ground track defined by its position  $(\lambda(t), \phi(t), h(t))$  at time  $t \in T$ . Satellites capture imagery at fixed time intervals (e.g., every second). At each time step, the satellite's coverage is characterized by its FoV center (in Equation 8) and the corresponding boundaries (in Equation 9).

We define a grid  $\Gamma$  as the constellation-level coverage, with a resolution of  $0.05^\circ$  in both longitude and latitude. While  $\Gamma(S, t)$  denotes the satellite-level coverage of satellite  $S$  at time  $t$ . Each element  $\gamma$  in  $\Gamma$  denotes the number of times that a specific grid point is observed by the satellite or the constellation. Thus, the primary objective is to optimize

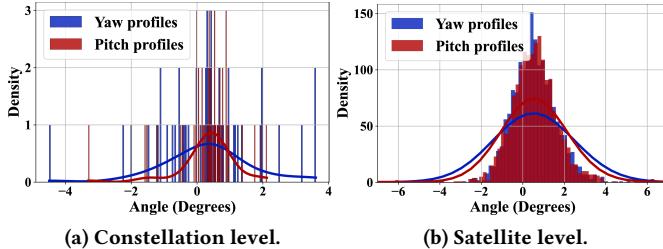


Figure 7: Attitude profile distribution of SpaceSched.

the constellation coverage, while avoiding over-observation and under-observation in some regions. It is specified by the grid coverage ratio  $CR$ , which quantifies the proportion of the grid that has been observed at least once, defined as:

$$CR[\Gamma] = \frac{\sum_S \sum_i \sum_j 1(\gamma(i, j) \neq 0)}{\sum_i \sum_j 1}, \quad (10)$$

where  $\gamma(i, j)$  represents the number of observed times at grid point  $(i, j)$ . Specifically, the indicator  $\gamma(i, j)$  is updated based on whether the grid point falls within the satellite's coverage, as follows:

$$\gamma(i, j) = \begin{cases} \gamma(i, j) + 1, & \text{if } \gamma(i, j) \in \Gamma(S, t), \\ \gamma(i, j), & \text{otherwise.} \end{cases} \quad (11)$$

**Meteorological dynamics.** To address issues of meteorological dynamics, we retrieve meteorological data to identify heavily contaminated areas, following the methodology established in prior work [69]. Notably, we deliberately exclude illumination conditions in our consideration, since those satellites equipped with SAR can capture imagery regardless of illumination conditions, and nighttime observation may also be important data content. Rather than introducing weather contamination as an additional optimization objective, which may increase computational complexity, we instead devise a contamination matrix  $\Upsilon$  to systematically track contaminated areas across the grid  $\Gamma$ . Specifically, any grid element within contamination areas is flagged in  $v \in \Upsilon$ , which allows effectively de-prioritizing weather-contamination areas. Specifically, the indicator  $\gamma(i, j)$  is updated as follows:

$$\gamma(i, j) = \begin{cases} 0, & \text{if } v(i, j) = 1, \\ \gamma(i, j), & \text{otherwise.} \end{cases} \quad (12)$$

**Attitude variation intensity.** Given the high orbital velocity of satellites, short-term attitude changes are inefficient and impractical. To balance the constellation coverage and energy efficiency, we design a step size  $\epsilon = T/t_s$ , where  $t_s$  represents the interval during which the satellite's attitude is held constant. The goal is to minimize the attitude variation intensity of the satellite, formulated as follows:

$$AVI = \frac{1}{\epsilon_n} \sum_{\epsilon=1}^{\epsilon_n} |\Delta\psi(\epsilon)| + |\Delta\theta(\epsilon)| + |\Delta\xi(\epsilon)|, \quad (13)$$

### Algorithm 1: Algorithm of the coverage distributor.

```

Input: Satellite constellations  $SAT$ , Time period  $T$ , Time offset  $\Delta T$ 
Output: Optimized yaw profiles  $\hat{\psi}$  and pitch profiles  $\hat{\theta}$ 
// Updating previously iterated satellites
1 for each satellite  $S_i \in SAT$  do
| // Setting bounds
2    $lb \leftarrow [\min(\psi), \min(\theta)]$ ; // lower bounds
3    $ub \leftarrow [\max(\psi), \max(\theta)]$ ; // upper bounds
| // Update the coverage grid  $\Gamma$ 
4    $[\lambda, \phi, h] = \text{obtainGroundTrace}(S_i, T, \Delta T);$ 
5   for each time  $t$  in  $T$  do
| |  $t_{step} = \left\lfloor \frac{T(t)}{T/\epsilon} \right\rfloor + 1$ ; // current time steps
| |  $\psi \leftarrow \hat{\psi}\{i-1\}(t_{step}); \theta \leftarrow \hat{\theta}\{i-1\}(t_{step}); \xi \leftarrow 0;$ 
| |  $\Gamma \leftarrow \text{obtainCoverage}(\lambda(t), \phi(t), h(t), \psi, \theta, \xi);$ 
| end
10 end
| // Coverage distributor optimization
11  $\text{objectiveFcn} = @(\mathbf{x}) \text{optimizationCovDis}(\mathbf{x}, \dots);$ 
12  $[\text{optAngles}, \text{fval}] = \text{ga}(\text{objectiveFcn}, 2 \cdot \epsilon, [], [], [], lb, ub, [], \text{opts});$ 
13  $\hat{\psi}\{i\} = \text{optAngles}(1:\epsilon); \hat{\theta}\{i\} = \text{optAngles}(\epsilon+1:end);$ 
| // Fitness function
14 Function fitness= $\text{optimizationCovDis}(\mathbf{x}, \dots)$ 
15  $\psi = \text{angles}(1 : \epsilon); \theta = \text{angles}(\epsilon + 1 : end);$ 
16 for each time  $t$  in  $T$  do
| |  $t_{step} = \left\lfloor \frac{T(t)}{T/\epsilon} \right\rfloor + 1$ ; // current time steps
| |  $\psi \leftarrow \psi(t_{step}); \theta \leftarrow \theta(t_{step}); \xi \leftarrow 0;$ 
| |  $\Gamma \leftarrow \text{obtainCoverage}(\lambda(t), \phi(t), h(t), \psi, \theta, \xi);$ 
| |  $\text{fitness} = -\alpha \cdot CR + \beta \cdot AVI$  // objective function
21 end

```

where  $\epsilon_n$  is the total number of step size,  $\Delta$  represents the angle variation over a given time interval  $t_s$ . As shown in Figure 7, the attitude profiles either at the constellation or satellite level render a value near zero. It is noted that, as discussed in § 4.2, the roll angle is fixed at zero due to its negligible impact on the coverage region.

**MOOP formulation.** Given the considerations outlined above, we formulate a Multi-Objective Optimization Problem (MOOP). Specifically, to achieve trade-offs in these competing objectives, we apply a weighted sum method [53] to form a unified optimization function, expressed as:

$$\min \left[ -\alpha \cdot \frac{\sum_S \sum_i \sum_j 1(\gamma(i, j) \neq 0 \text{ and } v(i, j) = 0)}{\sum_i \sum_j 1} \right. \\ \left. + \beta \cdot \frac{1}{\epsilon_n} \sum_{k=1}^{\epsilon_n} (|\Delta\psi(k)| + |\Delta\theta(k)| + |\Delta\xi(k)|) \right], \quad (14)$$

where  $\alpha$  and  $\beta$  weight the trade-off between maximizing constellation coverage while minimizing contamination region observations and attitude variation intensity.

**5.1.2 Increment-Accumulation Scheduling.** The existing optimization algorithms can be broadly categorized into exact algorithms (e.g., linear programming [14] and dynamic

programming [2]), heuristic algorithms (e.g., greedy algorithms [84]), and metaheuristic algorithms (e.g., evolutionary algorithms [9, 27]). Exact algorithms guarantee optimal solutions but fall short of being computationally infeasible; heuristic algorithms provide fast but often suboptimal solutions; while metaheuristic algorithms can balance exploration and exploitation to find near-optimal solutions efficiently in large search spaces. To this end, we adopt the Genetic Algorithm (GA) [27], a type of evolutionary algorithm, to solve the optimization problem. However, since satellites within the constellation are not fully engaged, optimizing constellation coverage with partial satellite availability becomes a challenge. Thus, *SpaceSched* leverages the principle of increment-accumulation, to independently assess and aggregate each satellite’s coverage contributions.

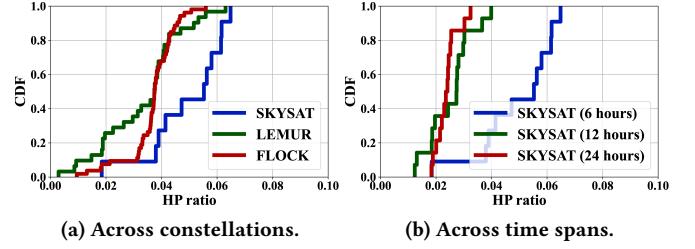
As shown in Algorithm 1, the initial population of candidate solutions is the attitude profiles fixed at zero for each satellite within the constellation, constrained within their predefined bounds. Each individual in the population is represented as a vector  $X_1$  of size  $2\epsilon$ , denoted as  $X_1 = [\psi_1, \psi_2, \dots, \psi_\epsilon, \theta_1, \theta_2, \dots, \theta_\epsilon]$ . The fitness function evaluates the constellation coverage throughout the time period  $T$ . *SpaceSched* optimizes the constellation coverage by iteratively adjusting the attitude profiles of each satellite, i.e., the principle of increment-accumulation. For each satellite, the previously iterated satellites with the assigned attitude profiles are evaluated in the context of the overall constellation, ensuring global constellation optimization. It is noted that the order in which satellites are considered has little impact on constellation coverage, because  $CR$  is proportionally small relative to the larger  $AVI$  penalties.

## 5.2 Satellite Selector

While the coverage distributor enhances constellation coverage by minimizing overlapping observations, it does not fully resolve ground track congestion due to inherent satellite count redundancy. The existing works typically choose the specific satellite group to enhance communication behaviors (e.g., RIS-assisted sub-THz access/relaying [26], satellite-ground contact [70], and inter-satellite link [44]), while *SpaceSched* refines the minimum set of satellites without compromising the attained service performance. However, ascertaining each individual satellite’s priority and its coverage sacrifices to the constellation is challenging. To this end, we propose a satellite selector to reduce the count of active satellites through combinatorial search.

**5.2.1 Hard-Soft Formulation.** Likewise, the satellite selector factors in several factors, including constellation ground tracks and coverage loss tolerance.

**Constellation ground tracks.** In contrast to the satellite-level ground track in the coverage distributor, the satellite



(a) Across constellations. (b) Across time spans.  
Figure 8: Illustration of the load fairness of *SpaceSched*.

selector operates at the constellation level by leveraging aggregated ground tracks. Given the constellation ground track tuple  $(\Lambda(T), \Phi(T), H(T))$  during  $T$ , a binary decision vector  $\mathbf{x} \in \{0, 1\}^n$  is employed, where each element  $x_i$  indicates whether satellite  $S_i$  is selected ( $x_{S_i} = 1$ ) or not ( $x_{S_i} = 0$ ). Consequently, the total satellite count is represented as  $\sum_{i=1}^n x_{S_i}$ .

**Coverage loss tolerance.** To ensure tolerant coverage loss while reducing satellite count, we set a coverage loss penalty function to maintain a soft coverage within a hard predefined threshold  $\tau$  constraint. Thus, the objective function of the satellite selector is formulated as:

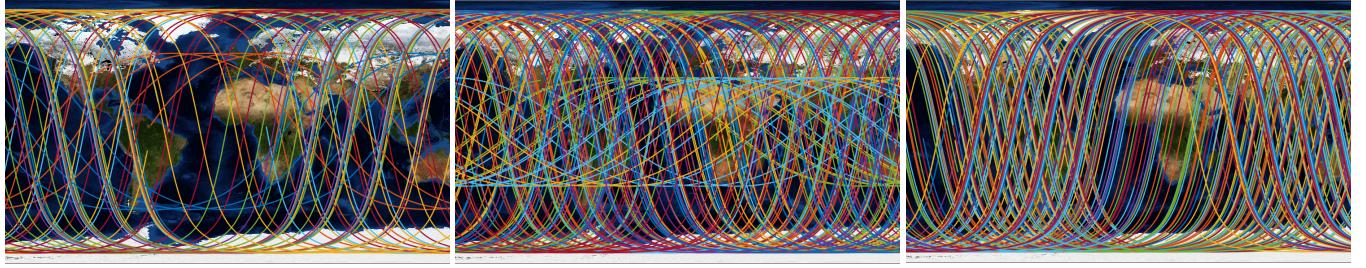
$$\min \begin{cases} \sum_{i=1}^n x_{S_i} + \eta \times |CR' - CR|, & \text{if } |CR' - CR| > \tau \\ \sum_{i=1}^n x_{S_i}, & \text{otherwise,} \end{cases} \quad (15)$$

where  $CR$  and  $CR'$  represent the coverage in the coverage distributor and the satellite selector, respectively; and  $\eta$  represents the penalty parameter in case of exceeding the predefined threshold  $\tau$ .

**5.2.2 Combinatorial Optimization.** Each candidate solution in the satellite selector is represented by a binary decision vector  $X_2$  of size  $n$  in  $SAT$ , denoted as  $X_2 = x_{S_1}, x_{S_2}, \dots, x_{S_n}$ , where  $x_{S_i} \in \{0, 1\}$  indicates whether satellite  $S_i$  is selected or not. The fitness function evaluates the selected satellite subsets with a soft tolerance to the constellation coverage from the coverage selector, as formulated in Equation 15. Crucially, each satellite is designed to possess an equitable priority in coverage contribution, as uneven utilization could lead to premature degradation of overused satellites while underutilizing others. As such, *SpaceSched* leverages the combinatorial search principle to identify the optimal satellite subset and enforces fairness across satellites (see § 6.2). Additionally, as for these unselected satellites, *SpaceSched* allows for the assignment of auxiliary or other lower-priority tasks to them, thereby improving the overall constellation utilization.

## 6 IN-SPACE DOWNLINK SCHEDULING

Following on-ground constellation scheduling, satellites may still encounter delayed delivery of high-value data due to the massive data load. While the existing downlink scheduling works [12, 65] mainly target resource allocation



(a) SKYSAT constellation (17 satellites). (b) LEMUR constellation (50 satellites). (c) FLOCK constellation (126 satellites).

Figure 9: Illustration of the ground tracks of three satellite constellations during 6 hours on the Earth map under Mercator projection.

or collision mitigation, and in-orbit edge computing solutions [15, 16] perform task-specific data filtering for bandwidth optimization, *SpaceSched* aims to alleviate data imbalance at both intra-satellite and inter-satellite levels. However, locating high-value data from the satellite’s local perspective, and enforcing intra- and inter-satellite load fairness are extremely challenging. To address these challenges, we propose a queue regulator to prioritize high-value data traffic (§ 6.1) under a fair load manner (§ 6.2).

## 6.1 Queue Regulator

Each satellite typically operates with a limited local perspective, so high-value data cannot be identified without the constellation-level global context. To this end, the selected active satellites are provided with a global coverage map, together with optimized attitude profiles and necessary satellite operation commands, from ground stations. The global coverage map  $\Gamma$  records the optimized coverage of the entire satellite constellation  $SAT$ , which is compact in size given the constraints on payload capacity.

Specifically, the queue regulator designs a dual downlink traffic queue architecture: a high-priority queue (*HP*) and a low-priority queue (*LP*). The high-priority queue prioritizes data that significantly impacts coverage performance or fulfills critical task demands (§ 7.5), while the low-priority queue handles those with relatively lower significance. To objectively assess data priority, the queue regulator utilizes  $p_{25}$ , denoting the 25th percentile value of coverage ratio  $CR$  within  $\Gamma(S, t)$ , i.e., the coverage profile of satellite  $S$  at time  $t$ . This statistical threshold effectively distinguishes between adequately covered and under-covered regions, thereby guiding the data propensity to form an appropriate queue. Specifically, the data  $D(S, t)$  priority is classified as follows:

$$D(S, t) = \begin{cases} \text{HP}, & \text{if } CR[\Gamma(S, t)] < p_{25} \\ \text{LP}, & \text{if } CR[\Gamma(S, t)] \geq p_{25}. \end{cases} \quad (16)$$

Each satellite within the constellation possesses its unique priority queue and transmits the data asynchronously.

## 6.2 Load Fairness

While the queue regulator, supported by the global coverage map, enables data prioritization at the intra-satellite level, it may potentially lead to uneven queue load distribution at the inter-satellite level. To be specific, some satellites might be overloaded with excessive *HP* queues while others remain underutilized.

To investigate this potential imbalance, we quantify the *HP* proportion in Figure 8 across constellations and time spans. The distribution of the *HP* proportion exhibits a high degree of balance, as evidenced by the steep incline of the CDF curve. Furthermore, the *HP* proportion is really small and remains consistently below about 7%. These two results suggest that the load is fairly distributed without overburdening any single active satellite. This balanced distribution is primarily attributed to the satellite selector’s fairness-aware combinatorial optimization strategy, which elaborately selects the satellite in an equitable manner.

## 7 EVALUATION

### 7.1 Methodology

**Implementation.** We develop *SpaceSched* by modeling orbital mechanics, traffic dynamics, and attitude control behaviors of satellite constellations. Specifically, the orbital mechanics are derived from real-world TLE orbit descriptors obtained from CelesTrak under Keplerian Laws [63]. The on-ground scheduling is executed on a workstation (with Intel Core i7-10700 2.90 GHz CPU, 64 GB RAM, and NVIDIA GeForce RTX 3080 GPU), while the in-space downlink scheduling is operated on Jetson TX2 on-board computing hardware (with 256-core NVIDIA Pascal™-family GPU and 8 GB RAM).

**Satellite constellations.** We evaluate *SpaceSched* on three large-scale, well-established satellite constellation systems: SKYSAT, LEMUR, and FLOCK, comprising 17, 50, and 126 evaluated satellites (derived from available and validated TLE files), as shown in Figure 9. Notably, the FLOCK constellation represents the world’s largest operational remote

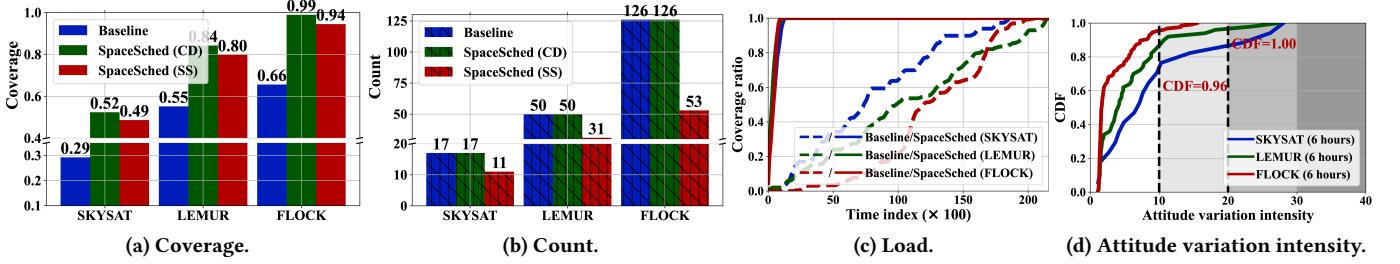
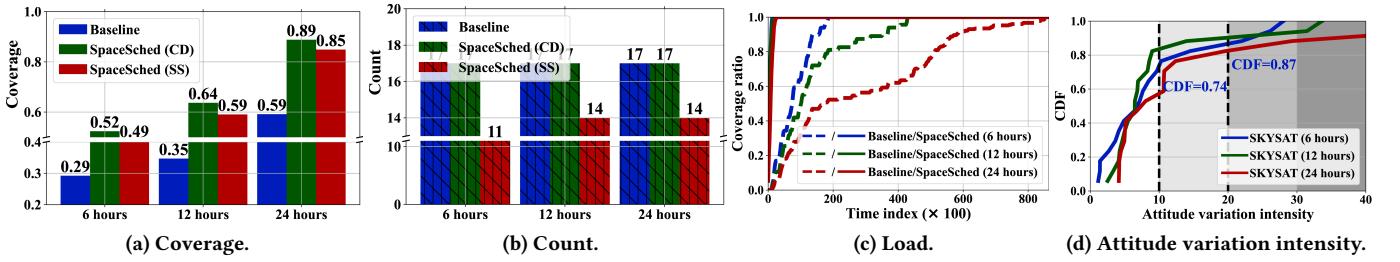
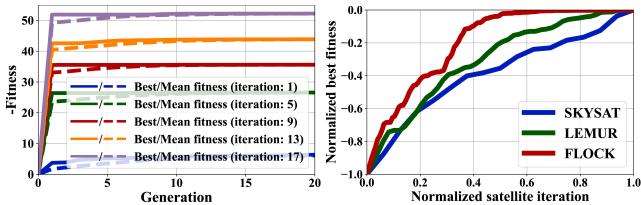
Figure 10: End-to-end performance of *SpaceSched* over three constellations.Figure 11: End-to-end performance of *SpaceSched* over three time spans.

Figure 12: Impact of generations. Figure 13: Impact of iterations.

sensing satellite constellation in scale to date [6]. The three constellations are detailed as follows:

- **SKYSAT** constellation consists of three satellite generations of A/B/C, operated by Planet Labs. SKYSAT delivers sub-meter resolution observation, multi-spectral, and video applications.
- **LEMUR** constellation, operated by Spire Global, provides a multi-use payload, such as remote sensing and GPS radio occultation.
- **FLOCK** is a fleet of nanosatellites (CubeSats), each with the size of a shoebox and weighing approximately 5 kg. FLOCK provides medium-resolution (3–5 meters) imagery with strong daily global coverage.

**Details.** The FoV is set to  $FoV_h = 7^\circ$  and  $FoV_v = 5^\circ$ . The observation range for evaluations is a geographically bounded range spanning a latitude of  $25\text{--}50^\circ$  and a longitude of  $-125\text{--}75^\circ$ , encompassing a significant portion of North America. Regarding operation parameters, the step size is  $\epsilon = 36$ , and the threshold is  $\tau = 0.5$ . The formulation parameters include  $\alpha = 100$ ,  $\beta = 0.05$ , and  $\eta = 1000$  as penalties and constraints. The optimization parameters are set to a generation of 20 and a population size of 100.

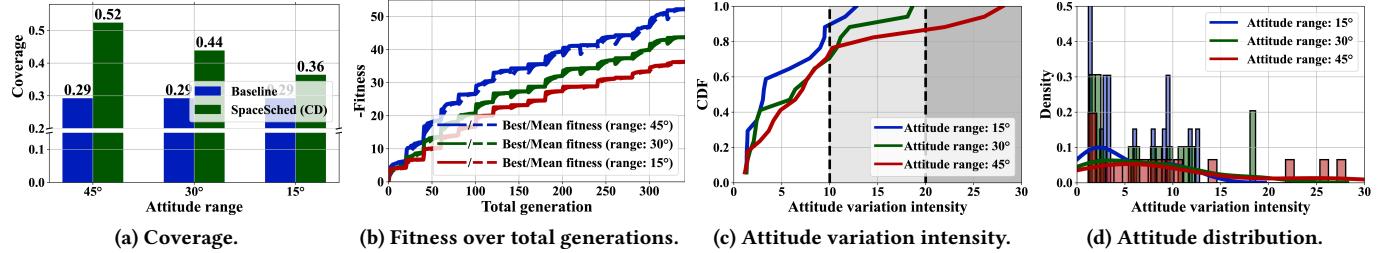
**Benchmarks.** We evaluate *SpaceSched* against *Baseline*, a plain satellite constellation configuration. Specifically, *Baseline* operates with the default Off Nadir Angle (ONA) (i.e.,  $0^\circ$ ), entire satellite set, and sequential downlink queue.

**Metrics.** We assess the following four performance metrics:

- (1) **Coverage.** The grid coverage ratio (Equation 9) quantifies the coverage performance of the coverage distributor.
- (2) **Count.** The active satellite count (Equation 15) measures the effectiveness of the satellite selector.
- (3) **Load.** The data ratio that achieves the expected coverage performance (i.e., 95%) serves as the data delivery load, to assess the queue regulator.
- (4) **Attitude variation intensity.** Mean sum of variations in satellite attitude profiles (Equation 13) characterizes the operational stability of satellites.

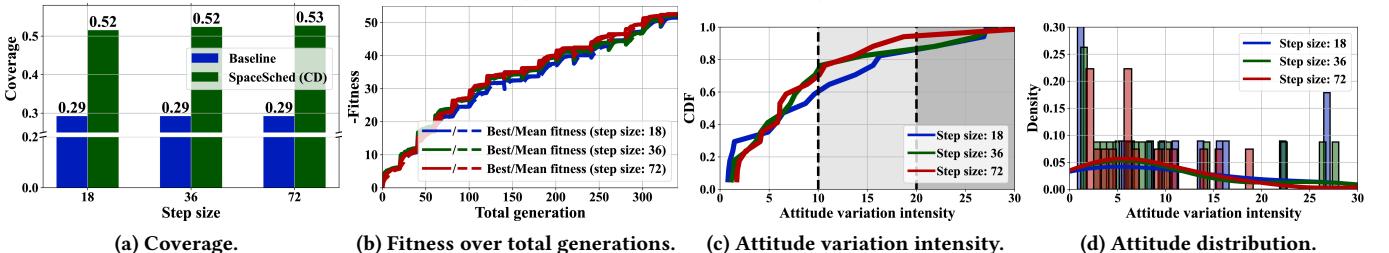
## 7.2 End-to-End Performance

We evaluate the end-to-end performance of *SpaceSched* on three satellite constellations of SKYSAT, LEMUR, and FLOCK, respectively. As illustrated in Figure 10, *SpaceSched* demonstrates a significant performance enhancement in coverage increment, count decrement, and load cut than *Baseline*. Specifically, *SpaceSched* achieves coverage of 52.38%, 84.24%, and 98.93%, outperforming *Baseline*'s 29.24%, 55.17%, and 65.63%, respectively. This is because a larger constellation typically provides greater flexibility in imagery capture. Furthermore, *SpaceSched* decreases the satellite count to 11, 31, and 53, compared to 17, 50, and 126 of *Baseline*. Conversely, satellites within a larger constellation typically share closer coverage regions, which suggests a more severe ground track congestion. At last, *SpaceSched* decreases the date delivery load to 5.09%, 4.17%, and 3.70%



(b) Fitness over total generations. (c) Attitude variation intensity.

Figure 14: Impact of attitude ranges.



(b) Fitness over total generations. (c) Attitude variation intensity.

Figure 15: Impact of step sizes  $\epsilon$ .

from *Baseline*'s 80.56%, 97.69%, and 85.65%. This suggests that *SpaceSched* can render an extremely small data load to ensure minimal delay latency. Overall, *SpaceSched* improves coverage by 1.51–1.79 $\times$ , reduces satellite counts by 1.55–2.38 $\times$ , and decreases data delivery load by 15.83–23.43 $\times$ , compared to *Baseline*.

Figure 11 presents the end-to-end performance of *SpaceSched* on the SKYSAT satellite constellation over three time spans of 6, 12, and 24 hours, respectively. Overall, *SpaceSched* improves coverage by 1.50–1.84 $\times$ , decreases satellite counts by 1.21–1.55 $\times$ , and decreases data delivery load by 15.83–36.46 $\times$ , compared to *Baseline*. Notably, as the time span increases, *SpaceSched* consistently improves larger coverage, necessitating more active satellites. However, the data load ratio declines due to incurring a greater overall transmission load over longer time spans, which indirectly suggests a more severe ground track congestion.

Furthermore, Figure 10d & Figure 11d illustrate the Cumulative Distribution Function (CDF) of the attitude variation intensity for *SpaceSched*. Overall, *SpaceSched* exhibits relatively low attitude variation intensity. In particular, for the SKYSAT constellation over the 6-hour span, the probability that the intensity is less than 20 is 86.67%, which is considered satisfactory for satellite operation.

### 7.3 Micro-benchmarking

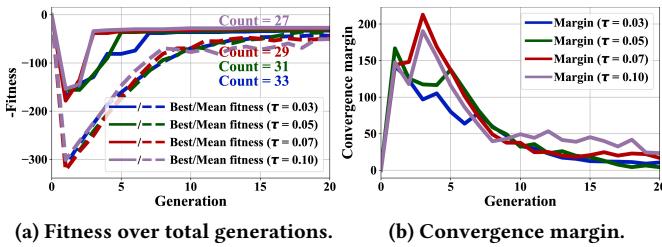
**7.3.1 Impact of Generation.** Figure 12 illustrates the fitness over generations of *SpaceSched* during the optimization process of the coverage distributor. As the generation increases, the best value of the fitness function shows a gradual improvement, with its mean value converging quickly to near-optimal solutions (about 10 generations). This indicates that *SpaceSched* is able to refine the performance rapidly over

time, but with diminishing returns as it approaches convergence. Furthermore, as satellite iterations increase, the fitness continues to improve due to more participating satellites and a broader solution space.

**7.3.2 Impact of Satellite Iteration.** Figure 13 illustrates the fitness over iterations. For better visualization, we normalize both the best fitness values and the satellite iterations across different constellations. We observe that the fitness improves rapidly during the initial iterations and tends to stagnate or experience only marginal improvements in the later iterations. This behavior indicates the presence of diminishing returns as more satellites are involved, suggesting the presence of redundant satellites.

**7.3.3 Impact of Attitude Range.** We evaluate the impact of attitude ranges of 45°, 30°, and 15° of *SpaceSched* in Figure 14. As the attitude range decreases, the coverage shrinks, rendering 52.38%, 43.86%, and 36.44% for the attitude range of 45°, 30°, and 15°, respectively. Furthermore, a smaller attitude range may induce a smaller attitude variation intensity. The reason is that a wider range allows for broader coverage but more unstable variation. This suggests an elaborate choice in attitude range can balance the coverage and the attitude variation intensity.

**7.3.4 Impact of Step Size.** We evaluate the impact of step sizes  $\epsilon$  of 18, 36, and 72 of *SpaceSched* in Figure 15. As the step size increases, we observe a slight improvement in coverage (i.e., 51.54%, 52.38%, and 52.71%) due to the more flexible attitude resolution. Furthermore, a larger step size results in smaller attitude variation intensity, as it promotes



(a) Fitness over total generations. (b) Convergence margin.

Figure 16: Impact of thresholds  $\tau$ .

smoother transitions between attitude profiles. These results suggest that optimizing the step size can strike a balance between maximizing coverage and minimizing control overhead, providing a practical guideline for system design.

**7.3.5 Impact of Threshold.** We evaluate the impact of varying thresholds  $\tau$  of 0.03, 0.05, 0.07, and 0.1 of *SpaceSched* in Figure 16. A higher threshold  $\tau$  allows *SpaceSched* to achieve a lower satellite count by permitting greater tolerance for coverage loss. Specifically, for the LEMUR constellation, the satellite count becomes 33, 31, 29, and 27, respectively. Furthermore, we observe that the optimization process in the satellite selector initially incurs a significant penalty in the early generations but converges rapidly thereafter. These findings suggest the importance of balancing the trade-offs between satellite count and coverage loss.

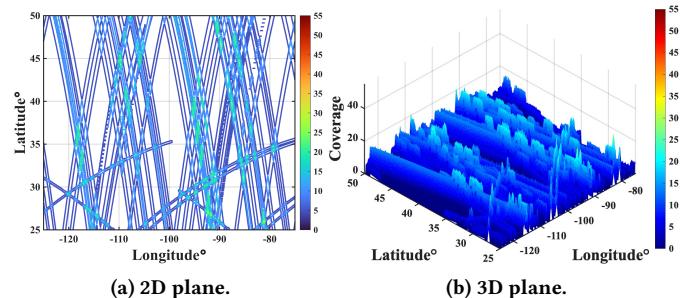
## 7.4 Quantitative Comparison

**Setup.** To further highlight the superior capabilities of *SpaceSched*, we conduct a quantitative comparison with the aforementioned benchmarks, i.e., the orbit-level organization methods. These kinds of methods require taking into account many limitations, such as policy regulation imposition and high-quality slot scarcity. Therefore, a fair and straightforward performance comparison is to select the same number of orbits (i.e., 31 as the satellite selector of *SpaceSched*) in the LEMUR constellation for comparison.

**Results.** Figure 9 illustrates the coverage of orbit-level organization methods, which only possess a ratio of 54.82%. It is significantly lower than 79.80% attained by *SpaceSched*, as illustrated in Figure 4. The primary reason is that the orbits are fixed prior to satellite launch, and cannot adapt to dynamic environments and task demands. While *SpaceSched* can dynamically adjust attitude over time to maximize satellite operations, which underscores the fundamental advantage of software-level solutions.

## 7.5 Case Study

**Setup.** To verify the *SpaceSched*'s potential to meet diverse remote sensing task demands, we deploy *SpaceSched* in different task demands, i.e., the spotlight mode. For the spotlight mode, the coverage distributor prioritizes maximizing



(a) 2D plane. (b) 3D plane.

Figure 17: Performance of the orbit-level organization methods.

the duration of observation on a given region. Thus, two metrics are used to evaluate the observation task: observation frequency (represented by the mean number of observations  $\mu_{\Gamma'}$  across  $\Gamma$ ) and observation uniformity (measured by the variance  $\sigma_{\Gamma'}^2$ ). A higher mean indicates frequent and sufficient observation, while a low variance suggests that the observations are well-distributed. With these demands, the optimization objective is represented by

$$\min [-\alpha' \times \mu_{\Gamma'} + \beta' \times \sigma_{\Gamma'}^2], \quad (17)$$

where  $\alpha'$  and  $\beta'$  are the weight parameters,  $\mu_{\Gamma'} = \frac{1}{IJ} \sum_{i=1}^I \sum_{j=1}^J \Gamma'_{ij}$ ,  $\sigma_{\Gamma'}^2 = \frac{1}{IJ} \sum_{i=1}^I \sum_{j=1}^J (\Gamma'_{ij} - \mu_{\Gamma'})^2$ . Furthermore, for the satellite selector, the threshold  $\tau$  is set to 5 as a coverage frequency loss tolerance. The optimization objective remains unchanged for the queue regulator.

**Results.** Figure 18 illustrates the performance of *SpaceSched* operating in spotlight mode. Overall, *SpaceSched* demonstrates a substantial improvement in coverage frequency and a decrement in satellite counts. Specifically, *SpaceSched* achieves a coverage frequency of 40.22 compared to 2.32 of *Baseline*, and reduces the satellite count to 9 from 17. While the downlink load remains smaller, even when the coverage frequency increases significantly. These results validate *SpaceSched*'s potential in various real-world applications, even in dynamic task adaptation to emergent situations.

## 8 DISCUSSION

**Imagery deformation.** While ideal imagery data is captured under ONAs, maintaining consistent ONAs over time proves challenging and operationally inefficient for satellites. In contrast, such flexibility provides opportunities for strategic attitude adjustments. Although attitude variations can introduce partial image distortions [77], *SpaceSched* alleviates these deformations by minimizing the attitude variation range and intensity (i.e., near-zero deformation levels as evidenced in Figure 7). Moreover, residual deformation can be further corrected using multi-channel data registration, geometric correction, and co-registration techniques.

**Latency, efficiency, and resilience.** *SpaceSched* operates through a coordinated ground-space architecture. Ground stations, with substantial processing capabilities, perform

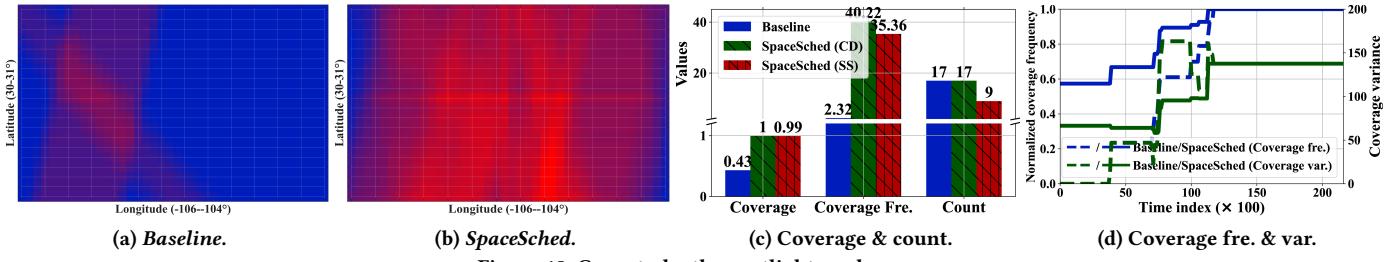


Figure 18: Case study: the spotlight mode.

computationally intensive optimization tasks in advance and periodically to render minimal latency. Satellites in space efficiently execute the received commands and perform the queue regulator. Furthermore, *SpaceSched* enables fault-tolerant operation through confidence-based restart capability, regardless of which side encounters an issue. Such responsive, efficient, and resilient design shows significant potential in real-world deployment.

## 9 RELATED WORK

**Satellite networking.** Recent advancements in satellite networking include but not limited to in-orbit edge computing [10, 15, 18, 49], ground station design [19, 25, 67, 72], networking protocols [8, 40, 43, 48, 61, 81], and security & privacy concerns [24, 50]. In-orbit edge computing has emerged to move the computation paradigm from various ground embedded devices [13, 23, 46, 55, 56, 60, 64, 75] to satellites. Specifically, Orbital Edge Computing (OEC) [16] is first proposed to organize satellite constellations into computational pipelines. Serval [69] relies on a bifurcated query execution that uses glacial filters to obtain relatively still boundaries from ground stations, and dynamic filters to query rapidly changing objects from satellites. Regarding ground station infrastructure design, PMSat [58] utilizes a phased array-coupled passive meta-surface to enhance satellite communication. Spectrumize [66] leverages the Doppler shift from satellites' motions as a signature to boost signal strength and de-multiplex transmission collisions. For networking protocols, SaTCP [7] is a cross-layer solution that exploits the predictability of satellite locations to prevent overly conservative Transmission Control Protocol (TCP) congestion. CosMAC [65] incorporates a constellation-scale medium access protocol for uplink, and a new network layer to schedule downlink traffic for Pico-satellites. SaTE [78] is a low-latency traffic engineering solution for dynamic LEO satellite constellations, which formulates a heterogeneous graph to infer traffic allocations. Additionally, plenty of works focus on GPS localization [28, 41] and GPS sensing [17].

**Satellite scheduling.** Satellite scheduling has attracted growing attention at both the satellite and constellation

scales. Early satellite-level scheduling works primarily pursue task-driven objectives, e.g., observation section, downlink transmission [39, 68]. At the constellation scale, scheduling becomes inherently multifaceted, involving system-wide service, load balancing, resource allocation, etc. Regarding the non-uniform reception of data across ground stations, Umbra [70] introduces a new scheduling algorithm to selectively under-utilize some links to ground stations. For one-to-many interference on ground stations, CosMAC [65] formulates downlink transmission as a graph problem to ensure enough interference-free contacts. Differently, *SpaceSched* focuses on resolving ground track congestion by the co-design of on-ground constellation scheduling and in-space downlink scheduling.

## 10 CONCLUSION

In this paper, we propose the design, implementation, and evaluation of *SpaceSched*, a hierarchical scheduling framework to resolve ground track congestion in satellite constellations. *SpaceSched* consists of two key components: an on-ground constellation scheduling pipeline, comprising a coverage distributor and a satellite selector, and an in-space downlink scheduling pipeline, featuring a queue regulator. Experimental results on three well-established satellite constellation systems demonstrate that *SpaceSched* improves coverage by up to 1.84×, reduces satellite counts by up to 2.38×, and decreases downlink queue load by up to 36.46×. Furthermore, our case study highlights *SpaceSched*'s potential to meet diverse task demands, which presents a transformative advancement for modern satellite constellations.

## ACKNOWLEDGMENTS

We sincerely thank the anonymous shepherd and reviewers for their constructive comments. This project was supported by the Innovation and Technology Commission of Hong Kong (Project No. MHP/072/23), the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. CityU 11202124 and CityU 11201422), and the NSF of Guangdong Province (Project No. 2024A1515010192).

## REFERENCES

- [1] Slingshot Aerospace. 2024. Congestion in Earth's orbit is getting even worse. <https://wwwaxios.com/2024/05/05/satellites-crowded-orbit-spacex>.
- [2] Richard Bellman. 1966. Dynamic programming. *science* 153, 3731 (1966), 34–37.
- [3] Pierre Bernhard, Marc Deschamps, and Georges Zaccour. 2023. Large satellite constellations and space debris: Exploratory analysis of strategic management of the space commons. *European Journal of Operational Research* 304, 3 (2023), 1140–1157.
- [4] David C Beste. 1978. Design of satellite constellations for optimal continuous coverage. *IEEE Trans. Aerospace Electron. Systems* 3 (1978), 466–473.
- [5] Aaron C Boley and Michael Byers. 2021. Satellite mega-constellations create risks in Low Earth Orbit, the atmosphere and on Earth. *Scientific Reports* 11, 1 (2021), 1–8.
- [6] Christopher Boshuizen, James Mason, Pete Klupar, and Shannon Spanhake. 2014. Results from the planet labs flock constellation. (2014).
- [7] Xuyang Cao and Xinyu Zhang. 2023. SaTCP: Link-Layer Informed TCP Adaptation for Highly Dynamic LEO Satellite Networks. In *IEEE INFOCOM 2023-IEEE Conference on Computer Communications*. IEEE, 1–10.
- [8] Jing Chen, Feilong Tang, Heteng Zhang, and Laurence T Yang. 2018. Lightweight retransmission for random access in satellite networks. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 549–557.
- [9] Yongliang Chen, Jinghui Zhong, Liang Feng, and Jun Zhang. 2019. An adaptive archive-based evolutionary framework for many-task optimization. *IEEE Transactions on Emerging Topics in Computational Intelligence* 4, 3 (2019), 369–384.
- [10] Zhuo Cheng, Bradley Denby, Kyle McCleary, and Brandon Lucia. 2024. EagleEye: Nanosatellite constellation design for high-coverage, high-resolution sensing. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 1*. 117–132.
- [11] Anthony J Correale. 2022. A Decision Analysis Framework to Consider Space Congestion in Orbit Selection. (2022).
- [12] Gaofeng Cui, Yanan Wang, Feng Li, and Weidong Wang. 2024. Completion Time Optimization with Coupled Uplink-Downlink Resource Allocation for Satellite Systems. *IEEE Trans. Aerospace Electron. Systems* (2024).
- [13] Kaiyan Cui, Qiang Yang, Yuanqing Zheng, and Jinsong Han. 2023. mmRipple: Communicating with mmWave radars through smartphone vibration. In *Proceedings of the 22nd International Conference on Information Processing in Sensor Networks*. 149–162.
- [14] George B Dantzig. 2002. Linear programming. *Operations research* 50, 1 (2002), 42–47.
- [15] Bradley Denby, Krishna Chintalapudi, Ranveer Chandra, Brandon Lucia, and Shadi Noghabi. 2023. Kodan: Addressing the computational bottleneck in space. In *Proceedings of the 28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 3*. 392–403.
- [16] Bradley Denby and Brandon Lucia. 2020. Orbital edge computing: Nanosatellite constellations as a new class of computer system. In *Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems*. 939–954.
- [17] Huixin Dong, Minhao Cui, Ning Wang, Lili Qiu, Jie Xiong, and Wei Wang. 2024. GPSense: Passive Sensing with Pervasive GPS Signals. In *Proceedings of the 30th Annual International Conference on Mobile Computing And Networking*. 1000–1014.
- [18] Kuntai Du, Yihua Cheng, Peder Olsen, Shadi Noghabi, and Junchen Jiang. 2025. Earth+: On-board satellite imagery compression leveraging historical earth observations. In *Proceedings of the 30th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 1*. 361–376.
- [19] Geneva Ecola, Bill Yen, Ana Banzer Morgado, Bodhi Priyantha, Ranveer Chandra, and Zerina Kapetanovic. 2025. SARLink: Satellite Backscatter Connectivity using Synthetic Aperture Radar. In *Proceedings of the 23rd ACM Conference on Embedded Networked Sensor Systems*. 398–410.
- [20] Christos N Efrem and Athanasios D Panagopoulos. 2020. Globally optimal selection of ground stations in satellite systems with site diversity. *IEEE Wireless Communications Letters* 9, 7 (2020), 1101–1104.
- [21] Austria European Space Policy Institute, Vienna and Ram S Jakhu. 2010. Iridium-Cosmos collision and its implications for space operations. *Yearbook on Space Policy 2008/2009: Setting New Trends* (2010), 254–275.
- [22] Cyrus Foster, Henry Hallam, and James Mason. 2015. Orbit determination and differential-drag control of Planet Labs CubeSat constellations. *arXiv preprint arXiv:1509.03270* (2015).
- [23] Ming Gao, Feng Lin, Weiye Xu, Muertikepu Nuermaimaiti, Jinsong Han, Wenyao Xu, and Kui Ren. 2020. Deaf-aid: Mobile IoT communication exploiting stealthy speaker-to-gyroscope channel. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*. 1–13.
- [24] Giacomo Giuliani, Tommaso Ciussani, Adrian Perrig, and Ankit Singla. 2021. {ICARUS}: Attacking low earth orbit satellite networks. In *2021 USENIX Annual Technical Conference (USENIX ATC 21)*. 317–331.
- [25] Chenwei Gu, Qian Wu, Zeqi Lai, Hewu Li, Yuxuan Weng, Weisen Liu, Jihao Li, Jun Liu, and Yuanjie Li. 2025. NovaPlan: An Efficient Plan of Renting Ground Stations for Emerging LEO Satellite Networks. In *IEEE INFOCOM 2025-IEEE Conference on Computer Communications*. IEEE, 1–10.
- [26] Sheikh Salman Hassan, Yu Min Park, Yan Kyaw Tun, Walid Saad, Zhu Han, and Choong Seon Hong. 2024. SpaceRIS: LEO satellite coverage maximization in 6G sub-THz networks by MAPPO DRL and whale optimization. *IEEE Journal on Selected Areas in Communications* (2024).
- [27] Chunlin He, Yong Zhang, Dunwei Gong, and Xinfang Ji. 2023. A review of surrogate-assisted evolutionary algorithms for expensive optimization problems. *Expert Systems with Applications* 217 (2023), 119495.
- [28] Songtao He, Mohammad Amin Sadeghi, Sanjay Chawla, Mohammad Alizadeh, Hari Balakrishnan, and Samuel Madden. 2021. Inferring high-resolution traffic accident risk maps based on satellite imagery and GPS trajectories. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 11977–11985.
- [29] Yvon Henri. 2020. The OneWeb satellite system. In *Handbook of Small Satellites: Technology, Design, Manufacture, Applications, Economics and Regulation*. Springer, 1091–1100.
- [30] Tao Huang, Zeru Fang, Qinjin Tang, Renchao Xie, Tianjiao Chen, and F Richard Yu. 2024. Dual-Timescales Optimization of Task Scheduling and Resource Slicing in Satellite-Terrestrial Edge Computing Networks. *IEEE Transactions on Mobile Computing* (2024).
- [31] Rasha H Ibrahim and Abdul-Rahman H Saleh. 2019. Re-Evaluation solution methods for Kepler's equation of an elliptical orbit. *Iraqi Journal of Science* (2019), 2269–2279.
- [32] Akira Iwasaki. 2011. Detection and estimation satellite attitude jitter using remote sensing imagery. *Advances in Spacecraft Technologies* 13 (2011), 257–272.
- [33] Zhong Ji, Yujin Liu, Chuanxi Zhao, Zhong Lin Wang, and Wenjie Mai. 2022. Perovskite Wide-Angle Field-Of-View Camera. *Advanced Materials* 34, 41 (2022), 2206957.

- [34] Erin Kahr, Oliver Montenbruck, and Kyle PG O'Keefe. 2013. Estimation and analysis of two-line elements for small satellites. *Journal of Spacecraft and Rockets* 50, 2 (2013), 433–439.
- [35] Patrick Kelly and Riccardo Bevilacqua. 2019. An optimized analytical solution for geostationary debris removal using solar sails. *Acta Astronautica* 162 (2019), 72–86.
- [36] Raymond Kristiansen and Per Johan Nicklasson. 2005. Satellite attitude control by quaternion-based backstepping. In *Proceedings of the 2005 American Control Conference, 2005*. IEEE, 907–912.
- [37] Enrico Lagona, Samuel Hilton, Andoh Afful, Alessandro Gardi, and Roberto Sabatini. 2022. Autonomous trajectory optimisation for intelligent satellite systems and space traffic management. *Acta Astronautica* 194 (2022), 185–201.
- [38] Lei Lei, Anyue Wang, Eva Lagunas, Xin Hu, Zhengquan Zhang, Zhiqiang Wei, and Symeon Chatzinotas. 2024. Spatial-temporal resource optimization for uneven-traffic LEO satellite systems: Beam pattern selection and user scheduling. *IEEE Journal on Selected Areas in Communications* (2024).
- [39] Michel Lemaitre, Gérard Verfaillie, Frank Jouhaud, Jean-Michel Lachiver, and Nicolas Bataille. 2002. Selecting and scheduling observations of agile satellites. *Aerospace Science and Technology* 6, 5 (2002), 367–381.
- [40] Jihao Li, Hewu Li, Zeqi Lai, Qian Wu, Yijie Liu, Qi Zhang, Yuanjie Li, and Jun Liu. 2024. Satguard: Concealing endless and bursty packet losses in leo satellite networks for delay-sensitive web applications. In *Proceedings of the ACM Web Conference 2024*, 3053–3063.
- [41] Ruinan Li, Xiaolong Zheng, Liang Liu, and Huadong Ma. 2024. Plug-and-play Indoor GPS Positioning System with the Assistance of Optically Transparent Metasurfaces. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*, 875–889.
- [42] Yijie Li, Weichong Ling, Taiting Lu, Yi-Chao Chen, Vaishnavi Ranganathan, Lili Qiu, and Jingxian Wang. 2025. MmBack: Clock-free Multi-Sensor Backscatter with Synchronous Acquisition and Multiplexing. *arXiv preprint arXiv:2507.01360* (2025).
- [43] Yuanjie Li, Lixin Liu, Hewu Li, Wei Liu, Yimei Chen, Wei Zhao, Jianping Wu, Qian Wu, Jun Liu, and Zeqi Lai. 2024. Stable Hierarchical Routing for Operational LEO Networks. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*, 296–311.
- [44] Yuanfeng Li, Qi Zhang, Haipeng Yao, Ran Gao, Xiangjun Xin, and F Richard Yu. 2024. Stigmergy and hierarchical learning for routing optimization in multi-domain collaborative satellite networks. *IEEE Journal on Selected Areas in Communications* (2024).
- [45] Zhiwei Li, Huanfeng Shen, Qihao Weng, Yuzhuo Zhang, Peng Dou, and Liangpei Zhang. 2022. Cloud and cloud shadow detection for optical satellite imagery: Features, algorithms, validation, and prospects. *ISPRS Journal of Photogrammetry and Remote Sensing* 188 (2022), 89–108.
- [46] Jianwei Liu, Yinghui He, Chaowei Xiao, Jinsong Han, Le Cheng, and Kui Ren. 2022. Physical-world attack towards WiFi-based behavior recognition. In *IEEE INFOCOM 2022-IEEE Conference on Computer Communications*. IEEE, 400–409.
- [47] Jie Liu, Bodhi Priyantha, Ted Hart, Yuzhe Jin, Woosuk Lee, Vijay Raghunathan, Heitor S Ramos, and Qiang Wang. 2015. CO-GPS: Energy efficient GPS sensing with cloud offloading. *IEEE Transactions on Mobile Computing* 15, 6 (2015), 1348–1361.
- [48] Lixin Liu, Yuanjie Li, Hewu Li, Jiabo Yang, Wei Liu, Jingyi Lan, Yufeng Wang, Jiarui Li, Jianping Wu, Qian Wu, et al. 2024. Democratizing {Direct-to-Cell} Low Earth Orbit Satellite Networks. In *21st USENIX Symposium on Networked Systems Design and Implementation (NSDI)* 24), 791–808.
- [49] Weisen Liu, Zeqi Lai, Qian Wu, Hewu Li, Qi Zhang, Zonglun Li, Yuanjie Li, and Jun Liu. 2024. In-Orbit Processing or Not? Sunlight-Aware Task Scheduling for Energy-Efficient Space Edge Computing Networks. In *IEEE INFOCOM 2024-IEEE Conference on Computer Communications*. IEEE, 881–890.
- [50] Wei Liu, Yuanjie Li, Hewu Li, Yimei Chen, Yufeng Wang, Jingyi Lan, Jianping Wu, Qian Wu, Jun Liu, and Zeqi Lai. 2024. The dark side of scale: Insecurity of direct-to-cell satellite mega-constellations. In *2024 IEEE Symposium on Security and Privacy (SP)*. IEEE, 445–464.
- [51] Sami Ma, Yi Ching Chou, Haoyuan Zhao, Long Chen, Xiaoqiang Ma, and Jiangchuan Liu. 2023. Network characteristics of LEO satellite constellations: A Starlink-based measurement from end users. In *IEEE INFOCOM 2023-IEEE Conference on Computer Communications*. IEEE, 1–10.
- [52] Sara Maqrot, Stéphanie Roussel, Gauthier Picard, and Cédric Pralet. 2022. Orbit slot allocation in earth observation constellations. In *PAIS 2022*. IOS Press, 3–16.
- [53] R Timothy Marler and Jasbir S Arora. 2010. The weighted sum method for multi-objective optimization: new insights. *Structural and multidisciplinary optimization* 41 (2010), 853–862.
- [54] Sujay Narayana, R Venkatesha Prasad, Vijay Rao, Luca Mottola, and T Venkata Prabhakar. 2020. Hummingbird: Energy efficient gps receiver for small satellites. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, 1–13.
- [55] Tao Ni, Zehua Sun, Mingda Han, Yaxiong Xie, Guohao Lan, Zhenjiang Li, Tao Gu, and Weitao Xu. 2024. Rehsense: Towards battery-free wireless sensing via radio frequency energy harvesting. In *Proceedings of the Twenty-Fifth International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing*, 211–220.
- [56] Jingyi Ning, Lei Xie, Zhihao Yan, Yanling Bu, and Jun Luo. 2024. Moirévision: A generalized moiré-based mechanism for 6-dof motion sensing. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*, 467–481.
- [57] Hiroki Nishiyama, Daigo Kudoh, Nei Kato, and Naoto Kadowaki. 2011. Load balancing and QoS provisioning based on congestion prediction for GEO/LEO hybrid satellite networks. *Proc. IEEE* 99, 11 (2011), 1998–2007.
- [58] Hao Pan, Lili Qiu, Bei Ouyang, Shicheng Zheng, Yongzhao Zhang, Yi-Chao Chen, and Guangtao Xue. 2023. PMSat: Optimizing Passive Metasurface for Low Earth Orbit Satellite Communication. In *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*, 1–15.
- [59] Darius Phiri, Matamyo Simwanda, Serajis Salekin, Vincent R Nyirenda, Yuji Murayama, and Manjula Ranagalage. 2020. Sentinel-2 data for land cover/use mapping: A review. *Remote Sensing* 12, 14 (2020), 2291.
- [60] Yidong Ren, Puyu Cai, Jinyan Jiang, Jialuo Du, and Zhichao Cao. 2023. Prism: High-throughput lora backscatter with non-linear chirps. In *IEEE INFOCOM 2023-IEEE Conference on Computer Communications*. IEEE, 1–10.
- [61] Yidong Ren, Amalinda Gamage, Li Liu, Mo Li, Shigang Chen, Younsuk Dong, and Zhichao Cao. 2024. SateRIoT: High-performance Ground-Space Networking for Rural IoT. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*, 755–769.
- [62] Christopher T Russell. 1971. Geophysical coordinate transformations. *Cosmic electrodynamics* 2, 2 (1971), 184–196.
- [63] John L Russell. 1964. Kepler's laws of planetary motion: 1609–1666. *The British journal for the history of science* 2, 1 (1964), 1–24.

- [64] Leming Shen, Qiang Yang, Kaiyan Cui, Yuanqing Zheng, Xiao-Yong Wei, Jianwei Liu, and Jinsong Han. 2024. Fedconv: A learning-on-model paradigm for heterogeneous federated clients. In *Proceedings of the 22nd Annual International Conference on Mobile Systems, Applications and Services*. 398–411.
- [65] Jayanth Shenoy, Om Chabra, Tusher Chakraborty, Suraj Jog, Deepak Vasisht, and Ranveer Chandra. 2024. CosMAC: Constellation-Aware Medium Access and Scheduling for IoT Satellites. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*. 724–739.
- [66] Vaibhav Singh, Tusher Chakraborty, Suraj Jog, Om Chabra, Deepak Vasisht, and Ranveer Chandra. 2024. Spectrumize: Spectrum-efficient Satellite Networks for the Internet of Things. In *USENIX Symposium on Networked Systems Design and Implementation, Santa Clara, CA*.
- [67] Vaibhav Singh, Akarsh Prabhakara, Diana Zhang, Osman Yağan, and Swarun Kumar. 2021. A community-driven approach to democratize access to satellite ground stations. In *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*. 1–14.
- [68] Panwadee Tangpattanakul, Nicolas Jozefowicz, and Pierre Lopez. 2015. A multi-objective local search heuristic for scheduling Earth observations taken by an agile satellite. *European Journal of Operational Research* 245, 2 (2015), 542–554.
- [69] Bill Tao, Om Chabra, Ishani Janveja, Indranil Gupta, and Deepak Vasisht. 2024. Known Knowns and Unknowns: Near-realtime Earth Observation Via Query Bifurcation in Serval. In *21st USENIX Symposium on Networked Systems Design and Implementation (NSDI 24)*. USENIX Association, 809–824.
- [70] Bill Tao, Maleeha Masood, Indranil Gupta, and Deepak Vasisht. 2023. Transmitting, fast and slow: Scheduling satellite traffic through space and time. In *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*. 1–15.
- [71] Deepak Vasisht and Ranveer Chandra. 2020. A distributed and hybrid ground station network for low earth orbit satellites. In *Proceedings of the 19th ACM Workshop on Hot Topics in Networks*. 190–196.
- [72] Deepak Vasisht, Jayanth Shenoy, and Ranveer Chandra. 2021. L2D2: Low latency distributed downlink for LEO satellites. In *Proceedings of the 2021 ACM SIGCOMM 2021 Conference*. 151–164.
- [73] Jerome R Vetter. 2007. Fifty years of orbit determination. *Johns Hopkins APL technical digest* 27, 3 (2007), 239.
- [74] Jingxian Wang, Junbo Zhang, Rajarshi Saha, Haojian Jin, and Swarun Kumar. 2019. Pushing the range limits of commercial passive {RFIDs}. In *In 16th USENIX Symposium on Networked Systems Design and Implementation (NSDI 19)*. 301–316.
- [75] Kun Wang, Jianqi Cao, Zimu Zhou, and Zhenjiang Li. 2024. Swapnet: Efficient swapping for dnn inference on edge ai devices beyond the memory budget. *IEEE Transactions on Mobile Computing* 23, 9 (2024), 8935–8950.
- [76] Darrel L Williams, Samuel Goward, and Terry Arvidson. 2006. Landsat. *Photogrammetric Engineering & Remote Sensing* 72, 10 (2006), 1171–1178.
- [77] Tim J Wright, Barry E Parsons, and Zhong Lu. 2004. Toward mapping surface deformation in three dimensions using InSAR. *Geophysical Research Letters* 31, 1 (2004).
- [78] Hao Wu, Yizhan Han, Mohit Rajpal, Qizhen Zhang, and Jingxian Wang. 2025. SaTE: Low-Latency Traffic Engineering for Satellite Networks. In *Proceedings of the ACM SIGCOMM 2025 Conference*. 896–916.
- [79] Haoran Xie, Yafeng Zhan, Guanming Zeng, and Xiaohan Pan. 2021. LEO mega-constellations for 6G global coverage: Challenges and opportunities. *IEEE Access* 9 (2021), 164223–164244.
- [80] Ruolin Xing, Mengwei Xu, Ao Zhou, Qing Li, Yiran Zhang, Feng Qian, and Shangguang Wang. 2024. Deciphering the enigma of satellite computing with cots devices: Measurement and analysis. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*. 420–435.
- [81] Yaoying Zhang, Qian Wu, Zeqi Lai, and Hewu Li. 2022. Enabling low-latency-capable satellite-ground topology for emerging leo satellite networks. In *IEEE INFOCOM 2022-IEEE Conference on Computer Communications*. IEEE, 1329–1338.
- [82] Mingmin Zhao, Peder Olsen, and Ranveer Chandra. 2023. Seeing through clouds in satellite images. *IEEE Transactions on Geoscience and Remote Sensing* 61 (2023), 1–16.
- [83] Qiang Zhao, Le Yu, Zhenrong Du, Dailiang Peng, Pengyu Hao, Yongguang Zhang, and Peng Gong. 2022. An overview of the applications of earth observation satellite data: impacts and future trends. *Remote Sensing* 14, 8 (2022), 1863.
- [84] ZiYan Zhao, MengChu Zhou, and ShiXin Liu. 2021. Iterated greedy algorithms for flow-shop scheduling problems: A tutorial. *IEEE Transactions on Automation Science and Engineering* 19, 3 (2021), 1941–1959.
- [85] Lipeng Zhu, Xiangyu Pi, Wenyan Ma, Zhenyu Xiao, and Rui Zhang. 2024. Dynamic beam coverage for satellite communications aided by movable-antenna array. *IEEE Transactions on Wireless Communications* (2024).