aerospace

MDPI

Article

A Three-Phase Progressive Multi-Beam Scheduling Algorithm for Large-Scale LEO Constellation TT&C Operations

Rongzhen Zhu [1,2], Yongqiang Li [1,2,*], Chenbin Wang [1], Xin Wen [1], Kunqi Li [1], Xiangyao Liu [1], Haibo Zhang [1] and Jichao Wang [1]

[1] School of Non-Commissioned Officer, Space Engineering University, Beijing 102206, China; zhuyue1848610482@163.com (R.Z.); 11212017@bjtu.edu.cn (C.W.); wx19951017@126.com (X.W.); calamansi1998@126.com (K.L.); 13324538698@163.com (X.L.); 18911938708@163.com (H.Z.); wangjichao@buaa.edu.cn (J.W.)

[2] Key Laboratory of Intelligent Space TTC&O (Space Engineering University), Ministry of Education, Beijing 101416, China

* Correspondence: hnlkzrz@126.com

Abstract

The multi-beam tracking, telemetry, and command (TT&C) scheduling problem for large-scale low Earth orbit (LEO) constellations carrying thousands of satellites brings formidable challenges. Strict resource limitations and visibility constraints trigger combinatorial explosion of feasible scheduling solutions. This paper proposes a three-phase progressive multi-beam scheduler (3PMS), which disassembles the complex integrated scheduling problem into hierarchically tractable subproblems. Phase I adopts a priority-aware first-come-first-served (FCFS) strategy combined with a dual-heap preemption mechanism to guarantee the execution of emergency tasks. Phase II implements load-balanced beam allocation based on a load-balance scoring function. Phase III introduces optimal execution window selection within visible arcs. Experiments are performed on two LEO constellation scenarios: the first is a single-shell orbital configuration with 1500 satellites, and the second is a three-shell architecture comprising 6080 satellites. Under extremely limited beam resources (1 beam), the worst-case single-beam capacity is 708 tasks per day, assuming every task consumes the maximum duration of 120 s, corresponding to 47.2% coverage for 1500 satellites and 11.6% for 6080 satellites. As the number of beams increases to five, all algorithms achieve 100% coverage for the 1500-satellite constellation, while the 6080-satellite constellation requires twenty beams for near-complete coverage. 3PMS demonstrates significant advantages in load balancing (Gini coefficient reduced from 0.0993 to 0.0003) and link quality (C/N improved by 5.5 dB). This paper verifies the feasibility of multi-beam scheduling algorithms for TT&C missions of large-scale satellite mega-constellations.

Keywords: LEO mega-constellations; multi-beam scheduling; fixed switching delay; load balancing; 3PMS algorithm

Check for updates

Academic Editor: Hyun-Ung Oh

Received: 20 July 2026
Revised: 29 August 2026
Accepted: 30 August 2026
Published: 3 September 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Recently, low Earth orbit (LEO) satellite constellations have entered an unprecedented era of rapid development, fundamentally reshaping the layout of global telecommunications, Earth observation and space-based services [1]. As of mid-2026, SpaceX's Starlink constellation has deployed more than 10,000 satellites, with follow-up expansion plans covering approximately 12,000–42,000 satellites [2]. OneWeb has completed its initial

deployment of 648 satellites, while China's GuoWang national satellite internet project intends to launch around 12,000 satellites across multiple orbital shells. Amazon's Project Kuiper, Telesat Lightspeed and a host of other commercial and governmental constellation projects are steadily advancing toward operational readiness. The global boom in mega-constellations represents a paradigm shift from traditional geostationary (GEO) and medium Earth orbit (MEO) architectures to distributed LEO networks [3,4], which deliver drastically reduced latency, enhanced throughout and full global coverage capability.

However, the exponential expansion of constellation scale imposes severe challenges on ground segment operations, particularly in tracking, telemetry and command (TT&C) management [5,6]. Traditional TT&C solutions that depend on manual scheduling, single-station single-satellite operation or simple rule-based heuristics are inherently unable to manage thousands of spacecraft running concurrently. The core difficulty stems from the severe imbalance between the exponentially rising number of satellites and the relatively slow expansion of ground-station resources, especially multi-beam antenna channels based on phased array antennas. Each LEO satellite only has a limited visibility window (normally 5–15 min per pass), during which it must complete all required TT&C operations, including tracking-based orbit determination, health telemetry downlink, and command uplink.

The TT&C scheduling problem for LEO satellite constellations must simultaneously address at least four interrelated constraints [7,8]: (1) satellite visibility constraints; (2) beam resource constraints; (3) link quality fluctuation; (4) emergency priority requirements. These intertwined constraints convert the original manageable operational planning task into an NP-hard combinatorial optimization problem with extremely high computational complexity. As constellation size expands from 100 s to 10,000 s of satellites, the solution space grows exponentially, rendering conventional scheduling algorithms computationally infeasible or generating severely degraded scheduling results.

With the rapid development of mega-LEO constellations, managing the massive resource demands at scale has received increasing attention. Over the past decades, extensive research has been conducted on satellite TT&C scheduling [7,9–12]. Gou et al. [13] proposed a three-tier hierarchical radio-resource management framework for mega-LEO constellations, addressing spectrum planning, beam scheduling, and user access at each tier. Mixed-integer linear programming (MILP) [14] and constraint programming (CP), as classical exact optimization methods, derive theoretically optimal solutions via exhaustive search within constraint boundaries. Barbulescu et al. [15] first applied MILP to NASA's Deep Space Network scheduling for small-scale scenarios, but exact algorithms are inherently limited by exponential complexity: they become computationally intractable for mega-constellations even with state-of-the-art solvers and decomposition techniques. Greedy heuristics and priority-based scheduling rules represent the most computationally efficient category, featuring complexity to support real-time operation, but suffer from limited global optimization perspective [16]. Standard greedy schemes also lack native support for task preemption and multi-objective trade-offs, restricting their use in modern differentiated TT&C systems. Metaheuristic algorithms [17] including genetic algorithms (GAs) have dominated satellite scheduling research for decades, balancing solution quality and efficiency for small-to-medium-scale problems through stochastic search mechanisms [18]. Their performance degrades sharply at mega-constellation scales due to the exponentially growing solution space and high parameter tuning overhead. Reinforcement learning (RL), especially deep reinforcement learning (DRL), has recently emerged as a popular approach for sequential scheduling tasks, with multiple studies reporting encouraging results on satellite scheduling benchmarks [19]. Hybrid frameworks integrating classical optimization [20], heuristics and machine learning represent the latest research trend [12,21,22], aiming to combine the complementary strengths of different

paradigms. Decomposition-based hybrid methods have shown scalability potential. Multi-beam scheduling is a well-studied subdomain within communication satellite systems [23], but communication-oriented methods cannot be directly applied to TT&C scenarios due to inherent differences in time scales and task properties, which necessitates dedicated algorithm design for TT&C resource scheduling. In 2025, Tang et al. [24] investigated the spacecraft operational task scheduling problem under constrained TT&C resources and introduced a task-splitting mechanism to improve ground-station utilization.

To address the multi-beam TT&C scheduling bottleneck for large-scale LEO constellations, this paper proposes a three-phase progressive scheduling algorithm (3PMS) based on hierarchical decomposition, which splits the strongly coupled NP-hard problem into three sequentially solvable subproblems with controllable complexity. Comparative experiments on two constellation scenarios verify that the algorithm maintains near-complete satellite coverage, runs much faster than classical genetic algorithms, and achieves 3.0–5.5 dB C/N gain and a 99.7% reduction in beam load Gini coefficient, effectively supporting real-time online scheduling of ultra-large LEO constellations. The rest of the paper is organized as follows. Section 2 formulates the scheduling problem with visibility pass, task and beam resource models, and defines optimization objectives and constraints. Section 3 details the 3PMS framework, covering dual-heap preemptive priority scheduling, load-balanced beam assignment, sliding-window intra-pass optimization, baseline algorithms and computational complexity analysis. Section 4 conducts multi-dimensional performance evaluation, supplemented by analysis on strategy comparison, parameter sensitivity and scalability. Section 5 summarizes the main conclusions and prospects future research directions. This work makes a valuable exploratory attempt at multi-beam TT&C scheduling for large-scale LEO constellations and offers technical support for the operation of next-generation ultra-large LEO constellations.

2. Problem Formulation

2.1. Visibility Pass Model

Satellite-ground visibility is modeled using pass segments generated by high-precision orbit propagation of a commercial software Satellite Tool Kit (STK). For satellite i relative to ground station, a visibility pass is defined as a tuple of time-series sampled parameters:

$$P_i = (t_{start}, t_{end}, Az(t), El(t), C/N(t), RxPower(t)) \quad (1)$$

where t_{start} and t_{end} denote pass start and end time; $Az(t)$ and $El(t)$ represent time-varying azimuth and elevation angles; $C/N(t)$ stands for carrier-to-noise ratio; and $RxPower(t)$ is received signal power. All parameters are sampled at a 10 s interval throughout the pass duration. Empirical analysis of two representative constellations reveals similar intra-pass characteristics, as shown in Table 1.

Crucially, both time-varying C/N and elevation angle follow distinct parabolic curves peaking near the midpoint of each pass, which aligns with the geometric laws of free-space path loss and antenna gain patterns. Approximately 98% of all passes have positive time slack, as the average pass duration of 539–592 s exceeds the standard 120 s task duration by 419–472 s. This abundant time redundancy forms the prerequisite for the intra-pass optimal interval selection in Phase III.

Table 1. Constellation configuration parameters.

Parameter	Constellation 1	Constellation 2
Total satellites	1500	6080
Orbital shells	1	3
Orbital planes	30	116
Altitude (km)	550	508–600
Inclination (deg)	53	50/55/85
C/N range (dB)	66–74	43–50
Total visibility passes	8761	40,349
Avg. pass duration (s)	539	592
Simulation period	24 h	24 h
Protection threshold T_prot (s)	60	60
Max. task duration (s)	120	120

2.2. Task Model

Each visibility pass generates at most one TT&C scheduling task. Task $\tau_{i,k}$, corresponding to the i-th pass of satellite k, is defined as:

$$\tau_{i,k} = (t_{arrival}, d_{task}, W_{feasible}, type, sat_id) \quad (2)$$

where $t_{arrival}$ is the arrival time (earliest executable start time); d_{task} is the required task duration (standard TT&C tasks require 120 s, with a minimum operational duration of 60 s; passes shorter than 60 s are excluded); $W_{feasible}$ is the feasible execution window; *type* is the task type (Regular/Emergency); and *sat_id* is the target satellite identifier. Emergency tasks are randomly generated with probability of 0.001, representing rare critical events such as anomaly response, collision avoidance maneuver commanding and emergency orbit adjustment. Each satellite is limited to at most three assigned passes per scheduling cycle to guarantee fair resource allocation across the entire constellation. The scheduling objective is to assign each task to a specific beam and a start time while satisfying the following rules: (a) no beam serves multiple tasks simultaneously; (b) beam switching constraints are fully complied; (c) emergency task preemption rules are strictly followed; (d) comprehensive indicators (coverage rate, link quality, scheduling efficiency) are optimized.

2.3. Beam Resource Model

The ground station is equipped with B independent multi-beam antenna channels sharing the same frequency band. Each beam can track and serve exactly one satellite at any moment. Although phased array antennas support ultrafast hardware-level beam steering at the microsecond scale, practical TT&C task switching still introduces non-negligible system-level delay dominated by signal reacquisition, carrier synchronization, and protocol handshaking. To characterize the practical switching overhead in actual TT&C operations, a fixed switching delay is adopted: T_{lock} = 2 s. The fixed switching delay of 2 s characterizes the unified system-level overhead for link reconstruction and state transition during beam task switching, which conforms to the inherent working mechanism of phased array multi-beam TT&C systems. This is the core advantage of phased array antennas over traditional mechanically scanned parabolic antennas. Constrained by mechanical inertia and servo response limitations, conventional parabolic antennas exhibit angle-dependent switching latency ranging from tens of seconds to minutes, which severely restricts the scheduling efficiency of high-dynamic LEO constellations.

2.4. Optimization Objectives and Constraints

The multi-beam satellite TT&C scheduling problem is modeled as a hierarchical constrained multi-objective optimization problem with distinct priority levels. In this work, a lexicographical optimization mechanism is adopted, and the primary optimization objective is satellite service coverage rate, defined as:

$$sat_rate = \frac{N_{served}}{N_{total}} \times 100\% \tag{3}$$

where N_{total} is the total satellite count within the constellation, and N_{served} denotes the number of satellites that complete valid TT&C scheduling tasks in a scheduling cycle. A full coverage rate of 100% is set as the highest-priority mandatory operational threshold, which must be satisfied for practical TT&C service deployment.

On the premise of guaranteed full coverage, three secondary lexicographical objectives are optimized to improve comprehensive scheduling performance, including link transmission quality, multi-beam resource allocation fairness, and emergency task response efficiency. To quantitatively evaluate beam load balance, the Gini coefficient is adopted, and its calculation formula is defined as:

$$G = \frac{2\sum(i \cdot x_i)}{n \cdot \sum x_i} - \frac{n+1}{n} \tag{4}$$

where n is the total number of multi-beam channels, and x_i represents the task load of the i-th beam. A smaller Gini coefficient indicates more uniform beam resource allocation. Specifically, the scheduling scheme aims to maximize the average C/N ratio of all assigned TT&C tasks, minimize the beam load Gini coefficient, and minimize the response latency of emergency TT&C tasks.

Hard constraints include: mutual exclusion (no simultaneous tasks per beam), window feasibility (execution within visible pass), legal preemption (regular tasks preemptable only after $\geq$60 s uninterrupted execution, denoted T_prot), per-satellite pass limit ($\leq$3 assignments), and execution continuity (2 s switching interval reserved between adjacent tasks).

3. Algorithm Design

The proposed three-phase multi-beam scheduler (3PMS) adopts a hierarchical decomposition framework to mitigate the intrinsic complexity of multi-beam TT&C scheduling via staged iterative refinement. To quantify the contribution of Phase III, the experimental evaluation also includes a reduced two-phase variant, denoted 2PMS (two-phase multi-beam scheduler), which executes Phases I and II only (dual-heap preemptive scheduling and load-balanced beam assignment); the complete three-phase algorithm 3PMS additionally applies Phase III (intra-pass window optimization).

In Phase I, Priority-Aware Task Sequencing and Dual-Heap Preemption Control, all arrival tasks are processed in chronological order; two independent priority min-heaps are maintained to separately manage emergency tasks (priority level 0) and conventional regular tasks (priority level 1) sorted by arrival time, idle beams are sequentially allocated to pending tasks, and resource conflicts are resolved through dedicated preemption rules, generating a primal feasible schedule that strictly guarantees the priority of emergency missions with a time complexity of $O(T\ log\ T)$, where T is the total task quantity and B denotes the number of multi-beam channels.

Phase II corresponds to Load-Balanced Beam Reallocation, in which a pure load-balance scoring function based solely on the current beam busy time is built to assess task-beam matching quality, and cross-beam task reassignment is performed to maximize

the load-balance score and equalize beam resource occupation, yielding an optimized scheduling scheme with relieved load imbalance at a complexity of $O(T \cdot B)$.

Phase III implements Intra-Pass Optimal Visible Interval Selection: for each task, a sliding-window enumeration strategy scans its full valid visible window to locate the optimal continuous sub-interval of fixed length *d_task* that maximizes combined link-quality metrics including C/N ratio and satellite elevation angle, and outputs the final near-optimal multi-beam TT&C schedule with a computational complexity of $O(T \cdot S)$, where S represents the average number of candidate visible windows per task and generally falls within the range of 40 to 50 in our simulation settings. The overall time complexity of the 3PMS algorithm is $O(T(S + B + log\ T))$; considering S and a beam count constraint B, the algorithm exhibits near-linear computational scalability, a critical property supporting efficient TT&C resource scheduling for large-scale LEO mega-constellations.

3.1. Phase I: Priority-Aware FCFS with Dual-Heap Preemption

Phase I constructs a dual min-heap data structure to establish independent priority queues for emergency tasks (priority 0) and regular tasks (priority 1), with all tasks sorted by arrival time to achieve priority-aware first-come-first-served scheduling. At each scheduling epoch, the algorithm extracts the highest-priority pending task from the non-empty heap queue. If idle beam resources are available, the task is assigned immediately to generate valid scheduling records. When all beams are fully occupied and the pending task is an emergency mission, the algorithm traverses all running regular tasks to screen eligible preemption objects. A regular task can be preempted only after achieving an uninterrupted execution duration of 60 s. Once a valid preemptable task is identified, the regular task is suspended and re-queued if residual execution time remains, while the emergency task seizes the beam resource for priority execution.

The preemption mechanism balances two core operational requirements: it ensures ultra-low response latency for sudden emergency missions through legal preemption, while protecting the execution integrity of conventional TT&C tasks via a 60 s protection threshold. This parameter conforms to the atomic execution characteristics of actual TT&C services, as most telemetry downloading and command injection processes require continuous and stable windows of 60–120 s to guarantee protocol validity and data reliability.

3.2. Phase II: Load-Balanced Beam Assignment

Based on the initial feasible schedule generated in Phase I, Phase II performs beam reallocation to improve load balancing across beams. The key observation is that FCFS assigns tasks to whichever beam happens to be free first, leading to unequal beam busy times and poor utilization of the overall beam pool. Phase II addresses this by computing a load-balance score for each candidate beam and redistributing tasks to more evenly share the workload. Each candidate beam is scored using the following load-balance metric:

$$Score(b) = 1 - (U_b \mid U_{max}) \tag{5}$$

In this equation, $Score(b)$ is the load-balance score for beam b. Here, U_b denotes the total busy time of beam b accumulated from Phase I, and U_{max} is the maximum busy time among all beams at that moment. Selecting b* = argmax $Score(b)$ therefore consistently directs tasks toward the currently least-loaded beam, achieving near-uniform workload distribution across all beams.

3.3. Phase III: Intra-Pass Sliding-Window Optimization

Aiming at the time-varying characteristics of C/N and elevation angle in satellite visible passes, this stage adopts a sliding-window traversal method to search for the optimal

execution time window for each TT&C task within the feasible visible arc. Given that the effective pass duration is far longer than the standard 120 s task duration, each pass contains dozens of optional continuous execution windows. This stage selects the optimal window with the highest comprehensive link quality to perform TT&C tasks, further improving the stability and reliability of satellite-ground data transmission.

To quantitatively evaluate window quality, this paper supplements the unified window evaluation function with min–max normalization:

$$WinEval = \alpha \cdot C/N_{win} + \beta \cdot El_{win} \quad (6)$$

where C/N_{win} and El_{win} represent the average carrier-to-noise ratio and average elevation angle of the candidate execution window, respectively. Both metrics are normalized to [0, 1] using min–max scaling: x_norm = (x − x_min)/(x_max − x_min), where x_min and x_max are the minimum and maximum values across all candidate windows for the same task. The weight coefficients α and β are fixed as 0.5 in this work, equally balancing link signal quality and geometric propagation superiority. This evaluation index fully considers the two core factors affecting TT&C link stability and transmission reliability.

3.4. Pseudocode and Algorithm Interpretation of the Proposed 3PMS

To demonstrate the hierarchical execution logic of the proposed three-phase progressive multi-beam scheduling (3PMS) algorithm, the standardized academic pseudocode is given in this section. Combined with the above theoretical design of the three stages, the pseudocode strictly restores the core mechanisms including dual-heap priority management, emergency task preemption rule, load-balanced beam matching scoring, and intra-pass sliding-window optimal screening. The overall algorithm takes satellite orbital pass data and TT&C task sets as input, iteratively optimizes the scheduling scheme through three progressive stages, and finally outputs the global optimal multi-beam scheduling result that satisfies all constraints and optimization objectives. The detailed pseudocode and step-by-step execution interpretation as shown in Algorithm 1.

Algorithm 1. Pseudocode of Three-Phase Progressive Multi-Beam Scheduling (3PMS).

Input: Orbital pass set P, TT&C task set τ, beam number B, switching delay T_lock = 2 s, protection threshold T_prot = 60 s
Output: Multi-beam scheduling scheme S

// Phase I: Priority-Aware FCFS with Dual-Heap Preemption
Initialize emergency_heap, routine_heap (min-heap by arrival_time)
For each task τ_i in τ (sorted by arrival_time):
    If τ_i.type == EMERGENCY: push to emergency_heap
    Else: push to routine_heap
While heaps not empty or beams busy:
    Release completed beams (record end Az/El)
    Advance clock to min(next_completion, next_arrival)
    Enqueue newly arrived tasks
    While emergency_heap not empty:
        b ← find_idle_or_preemptable_beam(T_prot = 60 s) // preempt routine after T_prot
        If b exists: assign task to beam b; else break

Algorithm 1. *Cont.*

```
12:   While routine_heap not empty:
13:     b ← find_idle_beam()
14:     If b exists: assign task to beam b; else break
15:     // Feasibility check: if remaining visible window < d_i, skip task (no truncation)

// Phase II: Load-Balanced Beam Assignment
16: For each scheduled task in S:
17:   If multiple idle beams available:
18:     score(b) = 1 − U_b/U_max   // pure load balance
19:     Select beam b* = argmax score(b)
20:     Reassign task to beam b*
21:     // Verify: no conflict with existing tasks on b* (mutual exclusion + T_lock separation)

// Phase III: Intra-Pass Sliding-Window Optimization
22: For each scheduled task τ_i with pass P_i:
23:   If duration(P_i) > d_task:
24:     For each candidate window [t, t + d_task] in P_i (step = 10 s):
25:       // Verify feasibility: window within visible pass, no beam conflict
26:       Evaluate WinEval = α × C/N_norm + β × El_norm   // α = β = 0.5, min-max normalized
27:     Select window with maximum WinEval
28:     Update task start_time and visibility_window
29: Return S
```

Algorithm Interpretation: In Phase I, the dual minimum heap structure realizes priority isolation and ordered scheduling of emergency and regular tasks. The 60 s execution protection threshold avoids frequent interruption of conventional TT&C services, which conforms to the continuity requirements of actual on-orbit measurement and control tasks, and effectively solves the conflict between emergency response capability and task execution integrity. In Phase II, based on the feasible initial scheduling scheme, the load-balance scoring function quantifies the utilization pressure of candidate beams. By maximizing the load-balance score, the algorithm breaks the local optimal limitation of time-sequence scheduling, achieves near-optimal beam load equalization, and improves the long-term operational stability of the multi-beam system. In Phase III, aiming at the parabolic variation law of elevation angle and C/N ratio in satellite visible passes, the sliding-window traversal mechanism fully excavates the time redundancy of effective passes. The window evaluation function comprehensively considers geometric propagation advantages and signal transmission quality, which further optimizes the reliability of TT&C data transmission on the premise of ensuring scheduling feasibility.

3.5. Computational Complexity Analysis

Table 2 intuitively compares the computational complexity growth trend of FCFS, Greedy + LS, GA, and the proposed 3PMS algorithm with the increase of task scale. Traditional metaheuristic algorithms such as GA exhibit polynomial complexity growth, while 3PMS maintains stable near-linear complexity, demonstrating outstanding scalability for mega-constellation scheduling tasks. The computational and space complexity of all comparison algorithms are systematically analyzed in this section. The overall time complexity of the proposed 3PMS algorithm is $O(T(S + B + log\ T))$, where S and B are fixed constants

independent of task scale T. Therefore, 3PMS achieves near-linear computational complexity, which is fundamentally superior to the polynomial complexity of metaheuristic algorithms such as GA and Greedy + LS. With the continuous expansion of future mega-constellations (exceeding 30,000 satellites), the near-linear scalability of 3PMS ensures long-term practical applicability, while traditional intelligent algorithms face explosive growth in computational overhead and cannot support online real-time scheduling.

Table 2. Computational complexity comparison.

Algorithm	Time Complexity	Space Complexity	Scalability
FCFS	O(T log T)	O(T + B)	Linear
GA	O(P·G·T·B)	O(P·T)	Poor
Greedy + LS	O(T·B + L·T)	O(T + B)	Moderate
3PMS	O(T(S + B + log T))	O(T·S + B)	Near-linear

The complexity of baseline algorithms is listed as follows: FCFS runs in O(T log T) with linear scalability, where T is the total number of tasks; Genetic Algorithm (GA) has a complexity of O(P · G · T · B), where P is population size and G is generation count, suffering from poor scalability; and Greedy + LS has moderate scalability with O(T · B + L · T) complexity, where L is the local search iterations. The hierarchical 3PMS framework maintains stable near-linear time complexity O(T(S + B + log T)) in all phases, where S is the average number of candidate windows per task (typically 40–50) and B is the beam count. Since S and B are fixed constants independent of task scale T, 3PMS achieves near-linear computational complexity, making it extremely suitable for large-scale ultra-dense LEO constellation scheduling scenarios.

4. Experiments and Results Analysis

4.1. Experimental Setup and Constellation Configurations

All comparative experiments were conducted based on high-precision STK orbital simulation data, with a ground station geographically deployed in Beijing. Two multi-scale LEO constellation scenarios and five beam resource configurations (B = 1, 2, 5, 10, 20) were tested, covering extreme resource-constrained and sufficient-resource operational scenarios. All algorithms adopted unified task parameter settings: task duration in the range [60, 120] s, a fixed switching delay T_{lock} = 2 s, maximum single-cycle pass limit $N_{pass,k} \leq 3$, and a 24 h continuous scheduling simulation.

As shown in Table 1, constellation 1 is a single-layer LEO constellation with 1500 satellites, 30 orbital planes, an orbital altitude of 550 km and an inclination of 53°. It generates 8761 valid visibility passes within 24 h, with an average pass duration of 539 s and a C/N range of 66–74 dB. Constellation 2 is a three-layer hybrid large-scale constellation containing 6080 satellites, with 116 orbital planes, altitudes ranging from 508 km to 600 km and three inclination angles. It obtains 40,349 valid passes, with an average pass duration of 592 s and a C/N range of 43–50 dB.

4.2. Coverage Rate Analysis

Figure 1 presents the satellite TT&C coverage rate of all comparison algorithms under all beam configurations for both 1500-satellite and 6080-satellite constellation scenarios. As a reference, we first derived the single-beam capacity within the 24 h scheduling cycle. Each task occupies a beam for T_task seconds plus a fixed switching delay of T_lock = 2 s. Because task durations in our scenarios span [60, 120] s, the per-task beam occupation is at most 120 + 2 = 122 s. A single beam can therefore complete at most n_max = $\lfloor 86{,}400/(\text{T_task} + \text{T_lock}) \rfloor$ = $\lfloor 86{,}400/122 \rfloor$ = 708 tasks within a day. We emphasize that this is a conservative

worst-case bound: it holds under the assumption that every task consumes the maximum duration of 120 s. If tasks were shorter, a beam would be released earlier and could serve more than 708 tasks; hence 708 is a guaranteed single-beam capacity in our setting rather than a universal upper bound on the number of schedulable tasks. In the evaluated scenarios, the bound is in fact tight: task durations are generated as min(pass_duration, 120 s) with a 60 s lower bound, and 98.6% of the tasks in the 1500-satellite scenario (99.0% in the 6080-satellite scenario) exactly reach the 120 s maximum, with a mean task duration of 119.6–119.7 s. Accordingly, the corresponding coverage fractions (47.2% for 1500 satellites and 11.6% for 6080 satellites) represent worst-case coverage bounds. This analytical bound is verified by the simulation results: at B = 1, Greedy + LS and GA both complete exactly 708 tasks, precisely reaching the theoretical limit. FCFS completes 686 tasks (45.7%) for the 1500-satellite constellation due to queue-order-induced fragmentation, while 3PMS achieves 44.1% coverage with a significantly higher C/N ratio (+5.5 dB) through link-quality-aware window selection, trading coverage for link quality. For the ultra-large 6080-satellite constellation, all algorithms achieve approximately 11.6% coverage at B = 1, consistent with the theoretical bound.

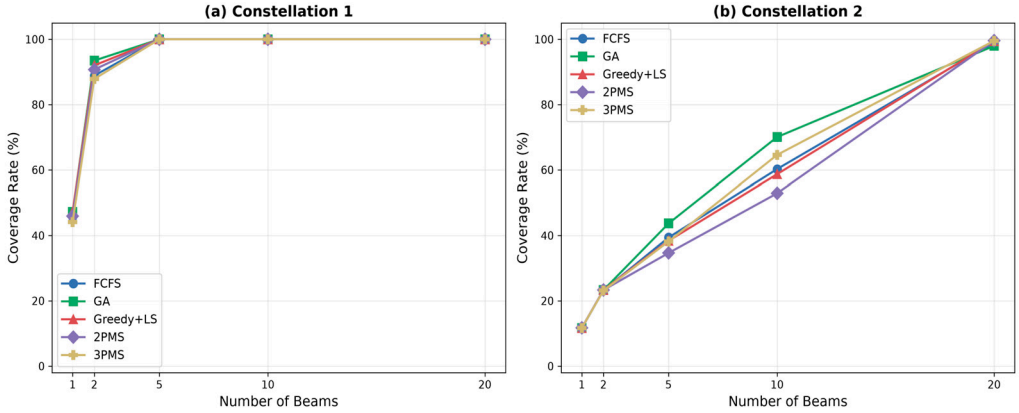


Figure 1. Coverage Rate Comparison.

As the number of beam resources increases, the coverage gap between algorithms gradually narrows. For the 1500-satellite constellation, all algorithms achieve 100% full coverage at B = 5. For the 6080-satellite constellation, coverage reaches approximately 60% at B = 10 (with GA achieving 70.1% through global search) and near-complete coverage (97.99–99.57%) at B = 20. After meeting the full coverage requirement, the performance competition shifts to secondary indicators including load balancing, link quality and scheduling efficiency.

4.3. Load-Balancing Analysis

Load balancing is a key index for long-term stable operation of multi-beam TT&C systems, which avoids long-term overload of individual beams and idle waste of partial resources, ensuring the stability and service life of phased array antenna equipment. Operationally, the Gini coefficient quantifies the inequality of task load distribution across beams: a value of 0 indicates perfectly uniform allocation (all beams carry equal load), while higher values reflect increasing concentration of tasks on fewer beams. In practice, a Gini coefficient below 0.01 indicates well-balanced load suitable for long-term continuous operation, whereas values above 0.1 signal significant imbalance that may lead to thermal overload of heavily loaded beams and accelerated equipment degradation. Figure 2 depicts the beam load Gini coefficient of all algorithms under different beam configurations. The two constellations exhibit fundamentally different load-balancing behaviors due to their distinct task-supply regimes. For constellation 1 (1500 satellites, max 4500 daily task-slots),

the 20-beam capacity of 14,160 slots far exceeds available tasks; FCFS sequentially assigns tasks to low-indexed beams, generating severe imbalance (Gini = 0.0993 at B = 10 and 0.4336 at B = 20). Phase II of 3PMS actively rebalances task-to-beam assignment, reducing Gini to 0.0003 at B = 10 and 0.0008 at B = 20—well below the 0.01 threshold. For constellation 2 (6080 satellites, max ~18,240 daily task-slots), the task demand exceeds 20-beam capacity (~14,160 slots), so all beams operate near saturation regardless of scheduling policy. Consequently, all algorithms—including FCFS—achieve Gini coefficients below 0.001, because load is naturally distributed across all available beams. In this regime, the load-balancing advantage of 3PMS is diminished, but this is not a weakness: both constellations benefit from the Phase II load balancer, and in the task-scarce regime (constellation 1), its contribution is decisive.

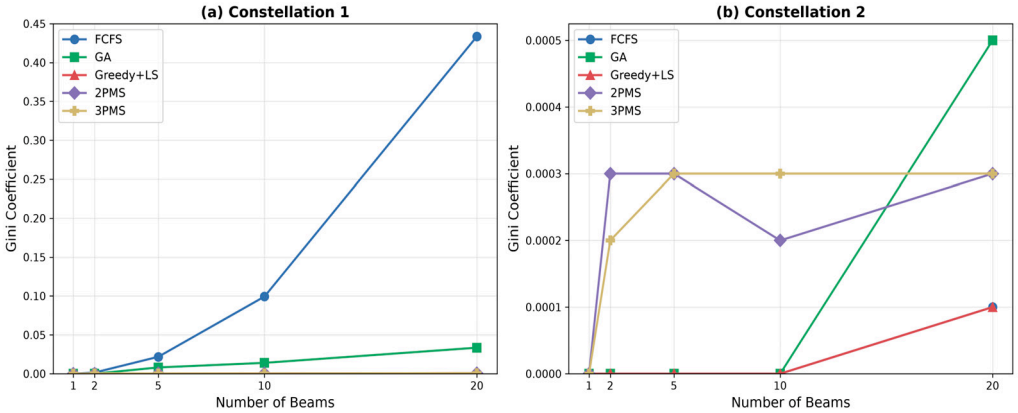


Figure 2. Load-balancing performance (Gini coefficient).

4.4. Link-Quality Analysis

Stable and high-quality star-ground link is the core premise for reliable TT&C data transmission, including telemetry downlink data integrity and command uplink accuracy. The proposed 3PMS introduces intra-pass sliding-window optimization, which screens the optimal execution window with high elevation angle and high C/N ratio for each TT&C task. As shown in Figure 3, simulation results show that compared with baseline algorithms, 3PMS achieves prominent link-quality improvement. For the 1500-satellite constellation, the C/N ratio of 3PMS reaches 73.9 dB at B = 1, which is 5.5 dB higher than the FCFS baseline (68.4 dB). This value approaches the upper bound of the overall C/N range of all visibility passes (66–74 dB as listed in Table 1) because the sliding-window optimizer specifically selects the sub-interval with peak C/N values within each pass, whereas the range in Table 1 represents the full variation across entire passes. For the 6080-satellite constellation with poorer overall link conditions (43–50 dB range), the C/N ratio is improved by up to 5.1 dB (48.5 dB vs. 43.4 dB). From the perspective of aerospace engineering, dB-level C/N optimization effectively increases the system link margin, reduces data bit error rate, and fundamentally improves the reliability of on-orbit TT&C missions.

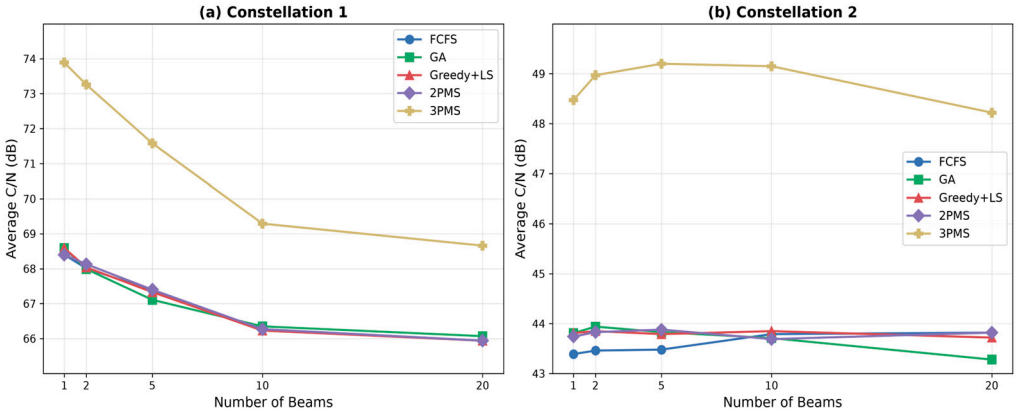


Figure 3. Link-quality comparison (average C/N ratio).

4.5. Beam Utilization Analysis

Beam resource utilization is a core engineering metric for evaluating the resource allocation efficiency of multi-beam TT&C systems. It directly characterizes the capacity to fully exploit finite ground-station antenna resources and mitigate idle resource waste. Figure 4 compares the beam utilization efficiency of various scheduling algorithms under different beam quantities. The theoretical maximum beam utilization is determined by the total task load and available beam-time: for the 1500-satellite constellation with max_passes = 3 (at $B \geq 5$), the upper bound is $1500 \times 3 \times 120/(B \times 86{,}400) \times 100\%$, yielding 62.5% at B = 10 and 31.3% at B = 20. The simulation results confirm that all algorithms operate within this theoretical limit.

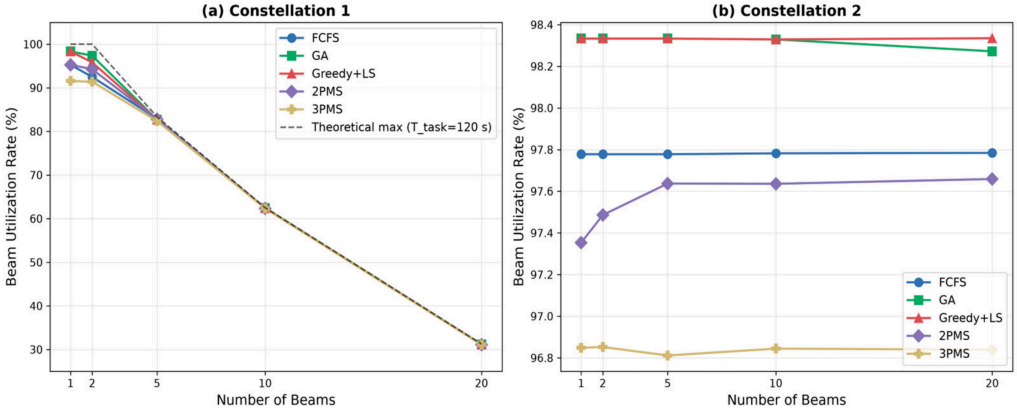


Figure 4. Beam utilization comparison.

Traditional scheduling algorithms merely adopt single time-sequence allocation logic and fail to balance task coverage assurance and resource utilization. In contrast, the hierarchical progressive optimization framework embedded in 3PMS effectively reconciles these two targets. Under the extreme single-beam constraint, all algorithms achieve saturated utilization above 91%, maximizing the scheduling capacity of the sole channel. At B = 10 for the 1500-satellite constellation, the beam utilization of all algorithms ranges from 62.3% to 62.5%, closely approaching the theoretical upper bound of 62.5%. At B = 20, utilization is approximately 31.1%, consistent with the theoretical limit of 31.3%. For the 6080-satellite constellation, beam utilization remains near 100% across all beam configurations due to abundant task demand. Such beam utilization performance stems from two core designs: the load-balanced beam assignment in Phase II and fine-grained sliding-window optimization in Phase III. These two modules enable precise matching between massive TT&C tasks

and limited beam resources, offering an efficient resource scheduling solution for long-term continuous operation of large-scale LEO constellation TT&C systems.

4.6. Computational Efficiency and Scalability Analysis

Real-time scheduling responsiveness and large-scale scalability are critical metrics for the online operation of mega-constellation TT&C systems. Simulation results demonstrate the superior computational efficiency of the proposed 3PMS algorithm under unified hardware and task parameter settings. For the medium-scale 1500-satellite constellation, the scheduling latency of 3PMS is restricted to 0.08–0.12 s, while the ultra-large 6080-satellite scenario yields a latency range of 0.41–0.76 s. In contrast, classic genetic algorithms require up to 28 s for iterative optimization under identical test conditions, rendering 3PMS up to about 350 times faster. Such a significant speed gap arises from the essential difference in algorithm paradigms: 3PMS adopts hierarchical greedy heuristics with heap-based sorting and local sliding-window adjustment, whereas genetic algorithms rely on multi-generation population iteration to search for near-optimal solutions, which inevitably introduces heavy computational overhead. Further scalability experiments validate its outstanding expansion capability. When the satellite scale expands by 4 times, the running time of 3PMS only rises by a factor of 5.1, demonstrating favorable near-linear scalability. Even for an extreme 42,000-satellite mega-constellation, the algorithm finishes all scheduling operations within 10 s, which fully satisfies the real-time online operation demands of large-scale LEO satellite systems.

Figure 5 illustrates the average first response time comparison across algorithms. The first response time is defined as the duration from a satellite's first visibility pass start to the time it receives its first successfully scheduled TT&C task; for all algorithms this metric is measured from the start of the first visibility pass, so the results are directly comparable. It reflects how quickly the scheduling system can initiate service for each satellite. The response time decreases monotonically with more beams for all algorithms, as additional beams reduce queuing delays. In constellation 1, the algorithms exhibit very similar response times when beam resources are scarce: at B = 1, 2PMS achieves 536.7 s, 3PMS 537.7 s, and FCFS 543.9 s. Differences emerge only at larger beam counts, where the window-placement choice of 3PMS takes effect: at B = 10, 3PMS shows the largest response time (372.5 s) while FCFS, 2PMS, Greedy + LS, and GA reach 252.9 s, 252.4 s, 225.9 s, and 208.6 s, respectively; at B = 20, 3PMS requires 202.3 s, whereas FCFS, Greedy + LS, and 2PMS drop to approximately 4 s and GA to 63.2 s. This is the price paid by 3PMS for its higher C/N: Phase III deliberately moves each task to the sub-window with the best link quality, which in most cases lies near the middle of the visibility pass, so the task starts later within its pass and its first response time grows accordingly. In constellation 2, where resource contention is substantially higher with 6080 satellites, all algorithms exhibit longer response times: at B = 10, 3PMS is again the slowest (609.2 s) versus 567.7 s (GA), 595.0 s (FCFS), 602.6 s (Greedy + LS), and 603.7 s (2PMS); at B = 20, 3PMS (571.9 s) trails GA (492.2 s), Greedy + LS (556.3 s), FCFS (557.4 s), and 2PMS (563.0 s). The convergence of FCFS, Greedy + LS, and 2PMS at B = 20 reflects their similar queue-ordering strategies under abundant beam resources, while GA's distinct optimization objective maintains a consistent speed advantage across configurations. The absolute response time values at B = 20 differ substantially between the two constellations: constellation 1 (1500 satellites, max 4500 daily task-slots) exhibits response times of only a few seconds at B = 20 because the 20-beam capacity (~14,160 slots) far exceeds the available tasks, so satellites are served almost at the very start of their visibility windows; in contrast, constellation 2 (6080 satellites, max ~18,240 task-slots) still shows non-trivial response times because task demand exceeds the 20-beam capacity. Both behaviors are fully consistent with the scheduling model.

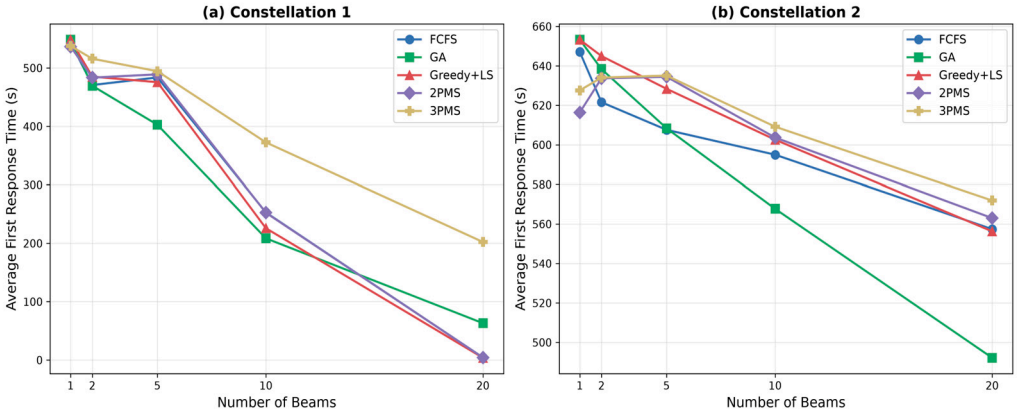


Figure 5. Average first response time comparison.

Figure 6 contrasts the performance of the proposed 3PMS algorithm across the two constellation configurations as the number of beams increases from 1 to 20. Panel (a) compares the coverage rate attained by 3PMS under the two constellations. In constellation 1, the algorithm achieves 44.1% coverage with only a single beam and reaches full (100%) coverage from B = 5 onward. In contrast, constellation 2—approximately four times larger—exhibits much stronger resource contention: with one beam only 11.7% of satellites are served, rising to 23.1% at B = 2, 38.2% at B = 5, 64.6% at B = 10, and 99.54% at B = 20. This demonstrates graceful scaling: a fourfold increase in satellite count requires roughly fourfold beam resources to achieve near-complete coverage, reflecting a near-linear and manageable growth in scheduling complexity. Panel (b) compares the C/N improvement of 3PMS over the FCFS baseline for the two constellations. The gain is consistently positive in both. In constellation 1 it declines from 5.49 dB at B = 1 to 2.72 dB at B = 20, remaining above 3.0 dB through B = 10; the larger gains at low beam counts reflect greater per-beam scheduling flexibility. In constellation 2 the gain is even more pronounced, varying between 4.40 dB (B = 20) and 5.72 dB (B = 5), confirming that the windowed link-quality optimization stays effective as satellite density grows. Notably, the C/N improvement tends to diminish with more beams, because the reduced per-beam load lowers the marginal benefit of link-quality-based selection. Nevertheless, the persistent 4.4–5.7 dB gain in the large 6080-satellite constellation verifies that the link-quality-aware windowed scheduling strategy scales to dense LEO constellations while preserving both coverage and link quality.

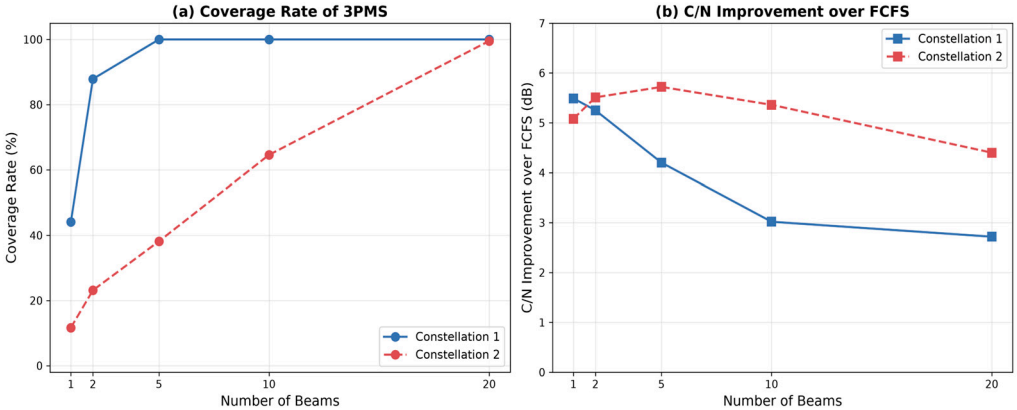


Figure 6. Scalability of 3PMS.

https://doi.org/10.3390/aerospace13090804

4.7. Discussion

Focusing on the core operational requirements of LEO constellation TT&C missions, the experimental results fully validate the practical value and technical advantages of the proposed 3PMS algorithm. Different from conventional communication-oriented satellite scheduling methods that merely pursue resource utilization efficiency, 3PMS prioritizes mission coverage integrity, on-orbit service fairness, stable link quality and real-time scheduling capability, which are the core concerns of aerospace TT&C engineering. The hierarchical progressive optimization framework effectively solves the performance degradation and computational explosion problems of traditional metaheuristic algorithms in ultra-large-scale constellation scenarios. The near-complete coverage guarantee, balanced beam load distribution, high-quality link screening and near-linear computational scalability make 3PMS highly adaptable to phased array multi-beam TT&C systems. Limited to the scope of this study, this paper only conducts single-ground-station deterministic scheduling. Future research will focus on multi-station collaborative TT&C scheduling and intelligent adaptive scheduling for complex dynamic space scenarios.

Figure 7 presents a five-dimensional radar chart comparing all five algorithms—FCFS, GA, Greedy + LS, 2PMS, and 3PMS—at B = 10 beams for both constellations. The five axes represent the normalized scores of: coverage rate, beam utilization, C/N link quality, load balance (inverse Gini), and response speed (inverse first response time). All metrics are normalized to [0, 1] using absolute scaling (coverage and utilization divided by 100, C/N normalized by theoretical ranges of 60–80 dB for LEO and 40–55 dB for GW, load balance as $1 - \text{Gini}$, and response speed by cross-algorithm min–max scaling). For constellation 1, all algorithms achieve 100% coverage at B = 10, so the Coverage axis shows no differentiation. The key differentiators are C/N quality and response speed: 3PMS achieves the highest C/N (0.465 vs. 0.31–0.32 for others, owing to its 69.3 dB average versus 66.2–66.4 dB) but the lowest response speed (corresponding to 372.5 s versus 208.6–252.9 s for the other algorithms), reflecting the link-quality-versus-responsiveness trade-off of its Phase III window selection. FCFS exhibits the lowest load-balance score (0.901, Gini = 0.0993), while the other algorithms achieve near-perfect load balance (Gini $\leq$ 0.014). For constellation 2, GA achieves the highest raw coverage (70.1%, normalized 0.701), while 3PMS again dominates C/N quality (0.61, from 49.2 dB versus 43.7–43.9 dB) and, consistently with constellation 1, shows the lowest response speed (609.2 s versus 567.7–603.7 s for the other algorithms). All algorithms achieve near-perfect load balance (Gini $\leq$ 0.0005) in the GW scenario at B = 10, reflecting the abundant beam resources relative to scheduling density. The radar chart demonstrates that 3PMS provides the best link quality at the cost of a longer first response time, while GA maximizes raw coverage in resource-constrained scenarios. This trade-off originates from Phase III: instead of shifting tasks to earlier sub-windows, Phase III selects the sub-window with the highest C/N and elevation within each visibility pass; because these metrics peak near the middle of the pass, the selected execution window tends to be centered in the pass rather than at its start, which is exactly why 3PMS combines a C/N advantage with a longer first response time.

Experimental results fully verify the superiority and engineering practicability of the 3PMS algorithm. Traditional heuristic and metaheuristic algorithms suffer from performance degradation or computational explosion under ultra-large-scale constellations, while the hierarchical decomposition framework of 3PMS stably outputs high-quality near-optimal solutions. The simultaneous optimization of coverage rate, load balancing, link quality and response efficiency solves the practical bottleneck of multi-beam TT&C scheduling for mega-constellations. Meanwhile, this work only focuses on single-ground-station deterministic scheduling. Future research can be expanded to multi-station

collaborative scheduling, dynamic uncertain task scheduling and intelligent parameter optimization scenarios.

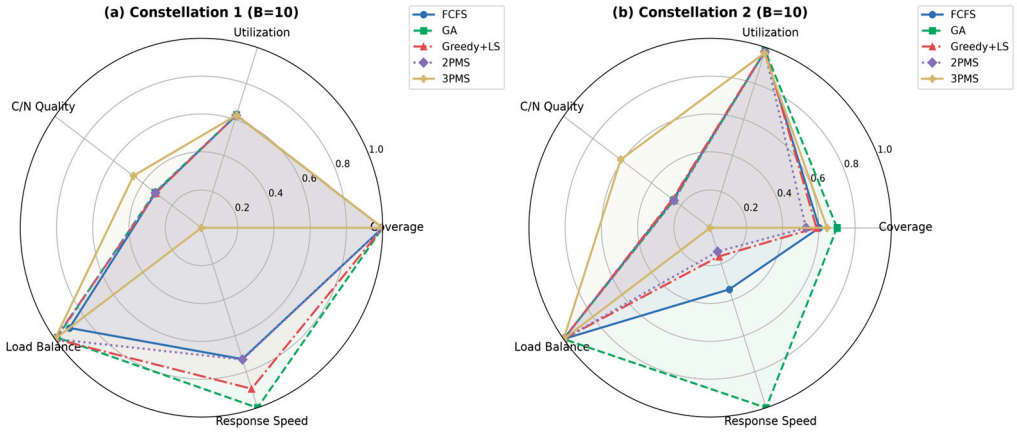


Figure 7. Multi-dimensional radar chart comparison at B = 10.

5. Conclusions

To address resource shortage and low scheduling efficiency in large-scale LEO mega-constellation TT&C systems, this paper proposes a three-phase progressive multi-beam scheduling algorithm (3PMS) tailored to phased array antennas. The method splits the NP-hard integrated scheduling problem into three hierarchical optimization modules: dual-heap preemptive priority scheduling, load-balanced beam assignment, and intra-pass optimal window selection. Simulations confirm the superior overall performance of 3PMS: (1) It achieves near-complete satellite coverage across medium and ultra-large constellations with favorable scalability—the single-beam worst-case bound of 708 tasks (under the assumption that every task lasts the maximum duration of 120 s) is verified, and full coverage is achieved with five beams for 1500 satellites and approximately twenty beams for 6080 satellites. (2) Its near-linear complexity delivers scheduling speed hundreds of times faster than genetic algorithms, enabling real-time online operation. (3) Joint optimization of beam load and satellite-ground links enhances the stability and fairness of TT&C resource allocation. (4) The framework fits the fixed switching delay property of phased arrays, showing strong engineering practicability.

This research offers an efficient, scalable scheduling scheme for mega-constellation TT&C resource management. Future work will explore multi-ground-station coordination, machine learning-driven adaptive scheduling, and exact benchmark solutions to boost the algorithm's generalization in complex space missions.

Author Contributions: Conceptualization, R.Z. and H.Z.; methodology, R.Z., X.W., X.L. and H.Z.; software, X.W.; validation, K.L. and J.W.; formal analysis, C.W. and K.L.; investigation, X.L.; resources, R.Z. and Y.L.; data curation, R.Z., C.W. and X.W.; writing—original draft preparation, R.Z. and K.L.; writing—review and editing, Y.L., X.L. and J.W.; supervision, Y.L.; project administration, Y.L. and H.Z.; funding acquisition, Y.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially supported by Key Laboratory of Intelligent Space TTC&O (Space Engineering University), Ministry of Education, No. CYK2025-01-10.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

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

References

1. Liu, Z.; Liu, J.; Liu, X.; Yan, J.; Cheng, Y.; Chen, Y. An iterated adaptive large neighborhood search algorithm for the large-scale communication satellite range scheduling problem. *Expert Syst. Appl.* **2025**, *278*, 127377. [CrossRef]
2. Jonathan's Space Pages Satellite Statistics: Satellite and Debris Population. Available online: https://planet4589.org/space/stats/active.html (accessed on 20 July 2026).
3. Cui, G.; Duan, P.; Xu, L.; Wang, W. Latency Optimization for Hybrid GEO–LEO Satellite-Assisted IoT Networks. *IEEE Internet Things J.* **2023**, *10*, 6286–6297. [CrossRef]
4. Liang, Q.; Fan, Y.; Yan, X.; Yan, Y. An algorithm based on differential evolution for satellite data transmission scheduling. *Int. J. Comput. Sci. Eng.* **2019**, *18*, 279. [CrossRef]
5. Li, C.; Xu, W.; Xu, L.; Wang, Y. An approach to multi-satellite TT&C resource scheduling based on multiagent technology and comprehensive weighted priority determination method. *J. Phys. Conf. Ser.* **2021**, *1812*, 012001. [CrossRef]
6. Zhan, Y.; Wan, P.; Jiang, C.; Pan, X.; Chen, X.; Guo, S. Challenges and Solutions for the Satellite Tracking, Telemetry, and Command System. *IEEE Wirel. Commun.* **2020**, *27*, 12–18. [CrossRef]
7. Zhan, Y.; Zeng, G.; Pan, X. Networked TT&C for Mega Satellite Constellations: A Security Perspective. *China Commun.* **2022**, *19*, 58–76. [CrossRef]
8. Xie, H.; Zhan, Y.; Lu, J. Mega-constellations based TT&C resource sharing: Keep reliable aeronautical communication in an emergency. *China Commun.* **2024**, *21*, 1–16. [CrossRef]
9. Wu, G.; Wang, T. Large-scale constellation TT&C resource scheduling algorithm based on adaptive simulated annealing. *Acta Aeronaut. Astronaut. Sin.* **2023**, *44*, 327759. [CrossRef]
10. Li, J.; Liu, J.-P.; Bai, J.; Yu, P.-J. The TT&C resource scheduling method based on multi-satellite. In Proceedings of the 2010 IEEE International Conference on Intelligent Computing and Intelligent Systems, Guilin, China, 22–24 October 2010; pp. 431–435.
11. Gao, J.; Han, H.; Li, H.; Li, Q.; Yang, Z.; Li, H. *A Large-Scale Constellation Distributed TT&C Mission Scheduling Method and Intelligent Automatic Management System*; Institute of Spacecraft System Engineer: Beijing, China, 2025.
12. Cui, W.; Zhang, Z.; Cao, X.; Zhao, X.; Sun, J. Intelligent planning method for large-scale low-orbit constellation supplementary network. *J. Phys. Conf. Ser.* **2021**, *1883*, 012123. [CrossRef]
13. Gou, L.; Bian, D.; Nie, Y.; Zhang, G.; Zhou, H.; Shi, Y.; Zhang, L. Hierarchical Resource Management for Mega-LEO Satellite Constellation. *Sensors* **2025**, *25*, 902. [CrossRef] [PubMed]
14. Kim, J.; Ahn, J.; Choi, H.; Cho, D. Task Scheduling of Agile Satellites with Transition Time and Stereoscopic Imaging Constraints. *J. Aerosp. Inform. Syst.* **2020**, *17*, 285–293. [CrossRef]
15. Barbulescu, L.; Watson, J.; Whitley, L.D.; Howe, A.E. Scheduling space–ground communications for the air force satellite control network. *J. Sched.* **2004**, *7*, 7–34. [CrossRef]
16. Lin, W.; Liao, D.; Liu, C.; Lee, Y. Daily imaging scheduling of an earth observation satellite. *IEEE Trans. Syst. Man Cybern.-Part A Syst. Hum.* **2005**, *35*, 213–223. [CrossRef]
17. Chen, M.; Wen, J.; Song, Y.; Xing, L.; Chen, Y. A population perturbation and elimination strategy based genetic algorithm for multi-satellite TT&C scheduling problem. *Swarm Evol. Comput.* **2021**, *65*, 100912. [CrossRef]
18. Holland, J.H. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*; University of Michigan Press: Ann Arbor, MI, USA, 1992. [CrossRef]
19. Herrmann, A.; Schaub, H. Reinforcement learning for the agile earth-observing satellite scheduling problem. *IEEE Trans. Aerosp. Electron. Syst.* **2023**, *59*, 5235–5247. [CrossRef]
20. Lin, M.; Yang, Q.; Jiahao, Q.; Ming, X. Massive Constellation Measurement and Control Scheduling Based on Hybrid Simulated Annealing Genetic Algorithm. *J. Astronaut.* **2023**, *44*, 1757–1766. [CrossRef]
21. Chen, X.; Tian, T.; Dai, G.; Wang, M.; Song, Z.; Xing, L. Deep reinforcement learning-based resource allocation method for multi-satellite scheduling. *Comput. Oper. Res.* **2025**, *181*, 107088. [CrossRef]
22. Du, Y.; Xing, L.; Chen, Y. Satellite scheduling engine: The intelligent solver for future multi-satellite management. *Front. Eng. Manag.* **2022**, *9*, 683. [CrossRef]
23. Duong, T.Q.; Nguyen, L.D.; Bui, T.T.; Pham, K.D.; Karagiannidis, G.K. Machine Learning-Aided Real-Time Optimized Multibeam for 6G Integrated Satellite-Terrestrial Networks: Global Coverage for Mobile Services. *IEEE Netw.* **2023**, *37*, 86–93. [CrossRef]
24. Tang, J.; Hou, Y.; Wu, S.; Si, Z.; Xu, J.; Qi, C. An Approach for Spacecraft Operational Task Scheduling Considering Constrained Space–Ground TT&C Resources and Task Splitting. *Aerospace* **2025**, *12*, 1077. [CrossRef]

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