Copyright Undertaking

DESIGN, DEPLOYMENT AND ON-ORBIT SERVICING OPTIMIZATION OF LEO SATELLITE CONSTELLATIONS

PENG HAN

PhD

The Hong Kong Polytechnic University

This programme is jointly offered by The Hong Kong Polytechnic University and Harbin Institute of Technology

2026

The Hong Kong Polytechnic University

Department of Industrial and Systems Engineering

Harbin Institute of Technology

Department of Control Science and Engineering

Design, Deployment and On-Orbit Servicing Optimization of LEO Satellite Constellations

Peng Han

A thesis submitted in partial fulfilment of the requirements

for the degree of Doctor of Philosophy

October 2025

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

__________________________ (Signed)

Peng Han (Name of student)

Abstract

Low-Earth orbit (LEO) satellite constellations have become vital for global communications, navigation, and remote sensing due to their low latency and high communication bandwidth. However, current design and deployment methods mainly address global or non-continuous regional coverage, lacking solutions for complex continuous coverage needs in specific regions, such as those required for emergency response like local conflicts, emergency rescue. Additionally, lifecycle management and on-orbit servicing remain challenging, with issues like satellite failures and propellant depletion yet to be systematically resolved.

Consequently, how to achieve efficient and economical design, deployment, and long-term operation of regional coverage LEO constellations have become emerging research problems in aerospace systems engineering. In response to these challenges, this dissertation addresses key system optimization problems throughout the LEO constellation lifecycle, including regional coverage design, launch and deployment optimization, and on-orbit servicing mission planning. The research proposes practical system optimization models and solution frameworks, whose effectiveness and superiority are validated through comprehensive simulations. The primary research contributions are outlined as follows:

First, a constellation design method based on Repeat Ground Track (RGT) orbits is proposed to satisfy the continuous coverage requirements of ground regional targets. The circular convolution property between constellation pattern vectors and target visibility vectors is established, leading to a linear integer programming model for regional coverage constellations. This model is subsequently extended to a simultaneous orbit optimization and satellite deployment (SOOSD) model for multiple RGT orbits. To address the computational complexity arising from high-dimensional and sparse constraints in the SOOSD model, a two-stage optimization framework integrating differential evolution (DE) with integer programming solvers is developed, enabling collaborative optimization of orbital parameters and satellite deployment positions. Simulation results across multiple ground regions of varying geometries and locations demonstrate that the proposed method not only satisfies complex regional continuous coverage requirements but also significantly reduces the required satellite count, outperforming traditional symmetric constellation designs and direct solution approaches while exhibiting robust performance under varying communication elevation angles and other parameters.

Second, for the launch and deployment (L&D) optimization of LEO constellations with orbital transfer vehicles (OTVs), a bi-objective combinatorial optimization model is formulated, incorporating constraints such as OTV fuel capacity, satellite payload per launch, and operational limitations. A genetic algorithm with cluster and nearest sorting initialization and local search operators (CNS-LSGA) is developed, employing the ϵ-constraint method to systematically explore the problem's Pareto frontier. The proposed solution framework efficiently generates high-quality initial solutions within the feasible domain and enhances solution quality through localized search strategies. Extensive simulations and comparative analyses validate the significant advantages of the proposed methodology in terms of solution quality, robustness, and diversity, providing an efficient computational tool and quantitative benchmark for constellation L&D mission planning.

Finally, for on-orbit servicing (OOS) mission planning of LEO constellations under the service station-service satellite mode, a two-stage optimization framework is proposed. The framework decomposes the system mission planning problem into service station orbit design and service satellite mission scheduling. In the service station orbit design phase, a modified spectral clustering and nonlinear programming approach (MSC-NLP) is employed to achieve target satellite assignment and optimal orbit design for multiple service stations. An enhanced hybrid clustering metric based on constellation orbital characteristics is introduced, substantially reducing overall mission fuel consumption. In the service satellite scheduling phase, a genetic quantum algorithm (GQA) is utilized to solve the mixed-integer nonlinear programming model for service satellite scheduling, jointly optimizing the number of service satellite deployments and total fuel consumption. Case study simulations demonstrate that the proposed solution framework outperforms existing methodologies and is well-suited for large-scale OOS scenarios, exhibiting strong potential for engineering implementation.

The dissertation intends to contribute to the LEO constellation systems engineering and provides technical guidance for the efficient construction and sustainable operation of LEO constellations. The proposed methodologies and findings offer valuable decision-making insights and technical resources for practical engineering challenges related to the design, launch and deployment, and on-orbit servicing of future regional coverage constellations.

Keywords: Low-Earth Orbit constellation deign, Repeat ground track orbits, Launch and deployment optimization, On-orbit servicing mission planning

Publications arising from the thesis

[1] **Han Peng**, Li Chuanjiang, Huang Chao*, Huang Hailong, et al. LEO Internet satellite constellation design for regional civil aviation airways with multiple repeat ground tracks. IEEE Transactions on Aerospace and Electronic Systems, 2025, 61(2): 4088-4104.

[2] **Han Peng**, Huang Chao*, Li Chuanjiang*, Guo Yanning. Bi-objective optimization of orbital transfer vehicle-based launch and deployment process for LEO constellation. Chinese Journal of Aeronautics, 2025.

[3] **Han Peng**, Guo Yanning, Wang Pengyu, Li Chuanjiang*, et al. Optimal orbit design and mission scheduling for sun-synchronous orbit on-orbit refueling system. IEEE Transactions on Aerospace and Electronic Systems, 2023, 59(5): 4968-4983.

Acknowledgements

I wish to express my sincere gratitude to The Hong Kong Polytechnic University (PolyU). During more than two years of duration as a joint Ph.D. student, I was fortunate to receive generous financial support and a comfortable working environment. The vibrant academic atmosphere and abundant research resources at PolyU have been instrumental in broadening my research perspective and fostering my academic growth. This experience has been invaluable, significantly expanding my horizons and enriching my academic journey.

I want to thank my supervisor, Prof. Yang Lidong, for his invaluable help and support throughout this project. Prof. Yang has given me a lot of advice and support in terms of teaching assistant work. His insights and expertise in academic research also inspired me.

I am profoundly grateful to my supervisor, Prof. Huang Chao. I vividly recall the winter of 2022, when I had received multiple rejection letters and was on the verge of abandoning my plans for an external joint training program. It was Prof. Huang's prompt response that reignited my confidence, ultimately granting me the opportunity to study in Hong Kong. Without her dedication and support, my joint doctoral program would not have progressed so smoothly. Prof. Huang's high standards for academic writing and careful attention to detail have greatly benefited my research. Her guidance and support throughout my graduation and career search have also been invaluable. Her optimistic and positive attitude has also profoundly inspired me.

My sincere thanks go to Prof. Huang Hailong. His patient guidance and selfless help strengthened my commitment to research. Prof. Huang's rigorous academic standards and care for his students have deeply impressed me, and his teachings will stay with me as I move forward.

I am also grateful to my supervisor in HIT, Prof. Li Chuanjiang. His meticulous attitude and concern for students greatly benefited me during my research. Prof. Li gave me complete academic freedom and respect, and also supported me in any of my choices. His dedicated guidance and pursuit of excellence will continue to inspire me.

I would like to thank all my fellow students in the HOIS research group at PolyU. Their support and encouragement made my research journey feel less lonely. I learned a lot from every group discussion, and our friendly conversations after meals helped ease my anxiety. Everyone's enthusiasm and positivity motivated me to keep going. I sincerely wish them all success in their research and a bright future.

I am grateful to my girlfriend, Miss Zhi Hui. She helped me through the most difficult times of my Ph.D. and restored my confidence in my research. Her understanding and patience helped me find balance under pressure, her companionship made my research journey less lonely, and her love made me appreciate the beauty of life. Her optimism has deeply influenced me, teaching me to face challenges in life with a better attitude.

I would like to thank my parents. Their emotional support was my greatest motivation to complete my doctoral studies. Their selfless dedication and support have been my source of strength.

Finally, I dedicate this thesis to the memory of my grandfather, who passed away from cancer during my doctoral studies. May he rest in peace. I will always miss my grandfather.

Contents

1 Introduction **1**

1.1 Research Background and Motivation 1

1.2 Introduction of LEO Constellations 4

1.3 Introduction of Satellite Launch and Deployment Methods 8

1.4 Introduction of Satellite On-Orbit Servicing Methods 11

1.5 Research Aims and Significance 15

1.6 Thesis Structure 16

2 Literature Review **19**

2.1 Literature Review of LEO Constellation Design 19

2.2 Literature Review of Constellation Launch and Deployment Optimization 22

2.3 Literature Review of On-orbit Servicing Mission Planning 26

2.3.1 Classification of OOS Mission Planning Problems 26

2.3.2 “One-to-many” and “Many-to-many” OOS Mission Planning Problems 28

2.3.3 “Service Station-Service Satellite” OOS Mission Planning Problems 30

2.4 Summary of Research Gaps 32

3 RGT Orbit-based Regional Coverage LEO Constellation Design 34
3.1 Introduction 34
3.2 Satellite-Target Coverage Analysis based on RGT Orbits 36
3.2.1 Satellite Orbit Dynamics and Orbit Representation 36
3.2.2 Repeat Ground Track Orbit 38
3.2.3 Satellite-to-Target Communication Coverage Condition 40
3.2.4 Coverage Representation 42
3.2.5 Coverage Calculation Based on Circular Convolution 43
3.3 Regional Coverage Constellation Design Model Based on RGT Orbits . . 47
3.3.1 Satellite Deployment Model Based on a Single RGT Orbit 48
3.3.2 Simultaneous Orbit Optimization and Satellite Deployment Model Based on Multi-RGT Orbits 50
3.4 Two-Stage Solving Framework Combining Differential Evolution and Integer Programming Solver 52
3.4.1 Design of the Solving Framework 53
3.4.2 Differential Evolution-Based Bi-Level Optimization Method . . . 55
3.5 Simulation of Case Study 57
3.5.1 Scenario and Simulation Parameter Settings 58
3.5.2 Simulation Results and Analysis of Single-Fold Coverage for a Mid-Latitude Complex-Shaped Target 59
3.5.2.1 Constellation Design and Flight Test Results 59
3.5.2.2 Comparison with Other Constellation Design Methods . 64
3.5.2.3 Comparison with Other Algorithms 67
3.5.3 Simulation Results and Analysis of Single-Fold Coverage for a High-Latitude Target 69

3.5.4 Simulation Results and Analysis of 2-Fold Coverage for a Narrow Target 71
3.5.5 Comparison with Results in Existing Literature 75
3.6 Summary 79

4 Bi-Objective Optimization of OTV-Based Launch and Deployment Process for LEO Constellation 80

4.1 Introduction 80
4.2 Mission Scenario Description 82
4.3 Modeling of the Launch and Deployment Optimization Problem 85
4.3.1 OTV Orbit Maneuver Fuel Calculation Model 85
4.3.2 Bi-Objective Optimization Modeling for Constellation Launch and Deployment 87
4.3.2.1 Decision Variables 88
4.3.2.2 Optimization Objectives 89
4.3.2.3 Constraints 90
4.4 Solution Framework Combining the ϵ-Constraint Method and CNS-LSGA 92
4.4.1 Bi-Objective Optimization Framework Based on the ϵ-Constraint Method 92
4.4.2 CNS-LSGA Algorithm Framework 94
4.4.2.1 Decision Variable Encoding 94
4.4.2.2 Cluster and Nearest Sorting Initialization Mechanism . 95
4.4.2.3 Genetic Operators 99
4.4.2.4 Local Search Operators 100
4.4.2.5 Overall Process of CNS-LSGA 102

4.5 Simulation of Case Study 103
4.5.1 Results on Regional Coverage Single-RGT LEO Constellation . . 104
4.5.1.1 Orthogonal Experiment for CNS-LSGA 105
4.5.1.2 Ablation Experiment for CNS-LSGA 106
4.5.1.3 Comparative Experiment for CNS-LSGA 107
4.5.1.4 Bi-Objective Optimization Result Comparison with Other MOEAs 109
4.5.1.5 L&D Plan Result Visualization 111
4.5.1.6 Sensitivity Analysis of Orbital Maneuver Duration . . . 111
4.5.2 Results on Large-Scale Walker LEO Constellation 113
4.5.3 Results on Regional Coverage Multi-RGT LEO Constellation . . 115
4.6 Summary 119

5 Mission Planning for LEO Constellation On-Orbit Servicing Systems 120
5.1 Introduction 120
5.2 On-Orbit Servicing Mission Scenario Based on the “Service Station-Service Satellite” Mode 121
5.2.1 Overall On-Orbit Servicing Mission Procedure 122
5.2.2 Single On-Orbit Servicing Mission Procedure 124
5.2.3 Objectives of Orbital Design and Mission Scheduling for On-Orbit Servicing Systems 125
5.2.4 Fuel Cost Calculation for a Single Round-Trip Orbit Maneuver of a Service Satellite 126
5.3 Optimal Deployment Orbit Design of Service Stations 128
5.3.1 Relative Drift Strategy of Service Stations and Client Constellation 129

5.3.2 Client Satellite Assignment by Modified Spectral Clustering . . . 131
5.3.3 Multiple Service Station Orbital Design by Nonlinear Programming 135
5.4 Mission Scheduling of Service Satellites 137
5.4.1 Modeling the Mission Scheduling Problem for Service Satellites . 137
5.4.2 Solving the Mission Scheduling Problem with a Genetic Quantum Algorithm 139
5.5 Overall Procedure of Orbital Design and Mission Scheduling for OOS Systems. 143
5.6 Simulation of Case Study 144
5.6.1 Feasibility Verification in Mission Scenario of Multi-Regional Coverage Constellation 145
5.6.1.1 Result of CS Assignment and SS Optimal Orbit Design 146
5.6.1.2 Result of Service Satellite Mission Scheduling 147
5.6.2 Comparative Study Based on Small-Scale SSO Constellation Mission Scenario 151
5.6.2.1 Result of CS Assignment and SS Optimal Orbit Design 152
5.6.2.2 Result of Service Satellite Mission Scheduling 155
5.6.3 Validation of Effectiveness under Large-Scale SSO Constellation Mission Scenario 157
5.7 Summary 163

6 Conclusions and Suggestions for Future Research 165

List of Figures

1.1 Starlink constellation 2

1.2 OneWeb constellation 2

1.3 Schematic comparison of the three types of satellite launch methods. The black arc represents the launch and de-orbit process. The red arc represents the orbital maneuver process of the OTV. The red dashed arcs represent the de-orbit process of the OTV. The green dashed arc represents the satellite release and autonomous orbital maneuver process. 9

1.4 ION satellite carrier 11

1.5 Exotrail's spacevan 11

3.1 Comparison of RGT orbit and non-RGT orbit. 40

3.2 Diagram of coverage conditions of satellites for ground targets. 41

3.3 Satellite coverage features in the same RGT orbit and target coverage profile calculation. 44

3.4 Two-stage solving framework for the constellation design problem based on RGT orbit [113]. 54

3.5 Geographical distribution of the regional airway targets and their split point target representations. 59

3.6 Iteration process for a single run of DE. 60
3.7 Satellite distribution of constellation design results of Case 1. 61
3.8 Satellite connectivity status of eight flights at different instants in Case 1. 64
3.9 Satellite-to-airplane accessible timeline during the flying of the eight flights in Case 1. 65
3.10 Accessible satellite number count during the flying of the eight flights in Case 1. 65
3.11 Comparison results between the two-stage solution framework and direct-solving algorithms under different ε settings in Case 1. 68
3.12 Satellite distribution of constellation design results of Case 2. 71
3.13 Satellite-to-airplane accessible timeline during the flying of the four flights in Case 2. 72
3.14 Accessible satellite number count during the flying of the four flights in Case 2. 72
3.15 Satellite distribution of constellation design results of Case 3. 73
3.16 Satellite-to-airplane accessible timeline during the flying of the two flights in Case 3. 74
3.17 Accessible satellite number count during the flying of the two flights in Case 3. 75

4.1 OTV-based launch and deployment process. 84
4.2 Splitter-based solution representation method. 95
4.3 Overall process of CNS-LSGA. 102
4.4 Satellite distribution of different single-RGT constellations in the $\Omega - M$ frame. 104

4.5 Orthogonal experiment results of CNS-LSGA for tuning the best parameters. 105
4.6 CNS-LSGA ablation results across sub-problems. Crosses indicate theoretical values involving constraint violations, while dots denote feasible iteration outcomes. 107
4.7 Comparison of algorithmic performance across distinct test instances. . . 108
4.8 Pareto fronts generated by four algorithms across three constellation scales. 109
4.9 Visualized Pareto-optimal L&D strategies for the 58-satellite case. Each sub-figure depicts initial ECI orbital positions on the left and the $\Omega - M$ distribution on the right. Colored arrows connect satellites within the same launch, indicating their deployment order. 112
4.10 Pareto front comparison under different T_m of single-RGT regional coverage constellation with 58 satellites case. 113
4.11 Satellite distribution of the multi-shell Walker constellation. 114
4.12 Pareto front comparison under the multi-shell Walker constellation test case. 115
4.13 Pareto front comparison of the four algorithms under the multi-RGT regional coverage constellation test case. 117
4.14 Visualizations of Pareto-optimal L&D results with 16 launches under the multi-RGT regional coverage constellation test case. 118

5.1 On-orbit servicing mission scenario based on the "service station-service satellite" mode [137]. 123
5.2 Single servicing mission procedure for one service satellite departing from SS and returning [137]. 124
5.3 Optimal orbit design of SS based on CS assignment result. 129
5.4 Two cases of relative RAAN drift between SS and CS 130

5.5 Diagram of Graph Segmentation in Spectral Clustering. 132
5.6 Schematic diagram of cluster distance metric between any two client satellites. 133
5.7 Overall optimization framework of orbital design and mission scheduling for OOS systems [137]. 143
5.8 Distribution of CS clusters and orbit drift of SSs (72 CSs and 5 SSs scenario). 148
5.9 OOS mission operation timeline of each service satellite in each SS (72 CSs and 5 SSs scenario). 150
5.10 Fuel cost for each OOS mission obtained by service satellite mission scheduling (72 CSs and 5 SSs scenario). 151
5.11 Optimal value tuning of λ (20 CSs and 3 SSs scenario). 152
5.12 Distribution of CS clusters and orbit drift of SSs (20 CSs and 3 SSs scenario). 153
5.13 Estimated fuel costs of each OOR mission (20 CSs and 3 SSs scenario). . 154
5.14 Comparison of estimated fuel costs in different mission duration (20 CSs and 3 SSs scenario). 155
5.15 OOR mission operation timeline of each service satellite in each SS (20 CSs and 3 SSs scenario). 156
5.16 Fuel cost for each OOR mission obtained by service satellite mission scheduling (20 CSs and 3 SSs scenario). 157
5.17 Satellite distribution of the SSO constellation in the $h_x - h_y$ frame. 158
5.18 Optimal value tuning of λ (100 CSs and 6 SSs scenario). 159
5.19 Distribution of CS clusters and orbit drift of SSs (100 CSs and 6 SSs scenario). 160
5.20 OOS mission operation timeline of each service satellite in each SS (100 CSs and 6 SSs scenario). 161

5.21 Fuel cost for each OOS mission obtained by service satellite mission scheduling (100 CSs and 6 SSs scenario). 161

List of Tables

1.1 Summary of typical satellite constellations. 8

1.2 Summary of typical OOS missions. 14

3.1 Simulation parameter settings. 60

3.2 Constellation design results of Case 1 60

3.3 Settings of the eight flights in Case 1. 63

3.4 Comparison results of different constellation design methods under different ε in Case 1. 66

3.5 Constellation design results of Case 2. 70

3.6 Settings of the four flights in Case 2. 72

3.7 Constellation design results of Case 3. 73

3.8 Settings of the two flights in Case 3. 74

3.9 Case settings in [28]. 75

3.10 Result comparison of Case 1 in [28]. 76

3.11 Result comparison of Case 3 in [28]. 77

3.12 Result comparison of Case 5 in [28]. 77

3.13 Runtime comparison. 78

4.1 Parameters of the OTV, satellite and CNS-LSGA. 103

4.2 Single RGT-based regional coverage constellations parameters. 104
4.3 L&D result comparison of the four algorithms under three regional coverage constellations test case. 110
4.4 Configuration of the multi-shell Walker constellation test case. 114
4.5 Quantitative comparison of L&D results for the multi-shell Walker constellation. 115
4.6 L&D result comparison of the four algorithms under under multi-RGT regional coverage constellation test case 117

5.1 Parameter settings of service satellite and mission scenario for the three test scenarios. 145
5.2 Results of CS clustering and optimal orbit design of corresponding SSs (72 CSs and 5 SSs scenario). 147
5.3 Service satellite mission scheduling results by GQA (72 CSs and 5 SSs scenario). 149
5.4 Results of CS clustering and optimal orbit design of corresponding SSs (20 CSs and 3 SSs scenario). 152
5.5 Comparison of service satellite mission scheduling results (20 CSs and 3 SSs scenario). 155
5.6 Results of CS clustering and optimal orbit design of corresponding SSs (100 CSs and 6 SSs scenario). 159
5.7 Result comparison of the four algorithms (100 CSs and 6 SSs scenario). . 162
5.8 Comparison of service satellite mission scheduling results (100 CSs and 6 SSs scenario). 162

Chapter 1

Introduction

1.1 Research Background and Motivation

In recent years, a growing global demand for space-based applications like communication, navigation, and remote sensing has led to a dramatic increase in the number of satellites in orbit. To enhance services for ground users, operators have launched hundreds or even thousands of satellites into specific orbits, forming satellite constellations. A satellite constellation is a group of artificial satellites working in a coordinated manner. These satellites are strategically placed to maintain specific spatial and temporal relationships, enabling them to provide coverage for designated regions or the entire globe.

Among these, Low Earth Orbit (LEO) satellite constellations have emerged as a primary focus for the global aerospace community. Their popularity stems from advantages such as low latency, high transmission rates, and flexible networking. In addition, advancements in satellite miniaturization and reusable rocket technology have significantly reduced launch costs. Consequently, large-scale global constellations like Starlink (Figure 1.1) and OneWeb (Figure 1.2) have been deployed and are now in commercial operation. Following this trend, China has also begun constructing two major LEO internet constellations, Qianfan and Guowang, since 2024.

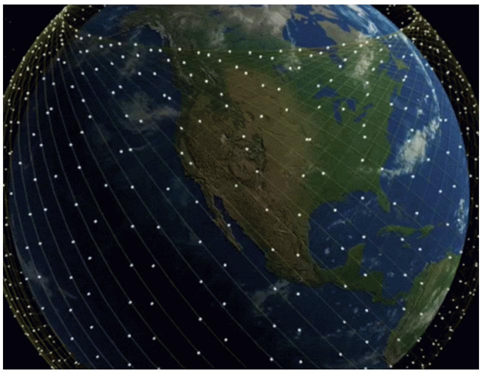


Figure 1.1: Starlink constellation. [1]

Figure 1.2: OneWeb constellation. [2]

However, for specific regional coverage requirements (such as national or regional communications enhancement, local conflicts, emergency disaster relief, etc.), the currently widely studied LEO constellations with global coverage capabilities face issues such as resource redundancy and difficulties in meeting complex requirements of specific regions. Systems like Starlink and OneWeb deploy thousands or even tens of thousands of satellites to ensure continuous global service. This wide-area design leads to resource waste when only a limited region requires coverage. Their uniform architecture also lacks flexibility for specialized regional demands. For example, providing high-speed satellite internet to civil aviation flights within a specific region involves coupled temporal and spatial coverage constraints that global constellations struggle to meet. In addition, global constellations can connect users anywhere, raising security and sovereignty concerns. They are unsuitable for tasks requiring strict confidentiality, such as emergency response or communications during local conflicts. Therefore, how to design regional

[1] https://tenor.com/zh-CN/view/spacexstarlink-gif-18203509

[2] https://spacenews.com/one-year-after-kickoff-oneweb-says-its-700-satellite-constellation-is-o

constellation architectures that meet complex coverage requirements at low cost has become a core challenge in the field of aerospace system optimization.

Furthermore, the deployment of LEO constellations faces several challenges. Complex constellation designs, diverse orbital requirements, limited rocket capacity, and rigid launch procedures can slow deployment, raise costs, and reduce mission success rates. For example, a launch service procurement for the Qianfan constellation failed in early 2025 because existing launch vehicles and procedures could not meet its complex launch and deployment needs [3]. As rocket capacity is unlikely to grow quickly, the conventional two-stage ”launch vehicle-satellite” model is becoming inadequate. A promising alternative is to use Orbital Transfer Vehicles (OTVs). These vehicles can perform multiple maneuvers in space to reach various deployment positions, forming a three-stage ”launch vehicle-OTV-satellite” architecture. This approach enables continuous, efficient, and rapid deployment of multiple satellites, but it also introduces new challenges for optimizing launch and deployment at scale.

Finally, the rapid development of LEO constellations is causing a significant increase in the number of satellites, making the LEO environment increasingly crowded. Once satellites run out of fuel or are decommissioned, they can no longer maintain their orbits or maneuver. As a result, they become space debris, which can take decades or even centuries to naturally deorbit and burn up in the atmosphere. These non-operational satellites occupy valuable orbital space and increase the collision risk for active satellites, posing a major threat to the space environment. Therefore, efficient on-orbit servicing (OOS) systems are needed to ensure the long-term safety and operation of constellations. These systems are designed for complex tasks such as refueling, repairs, and debris removal to prevent

[3] https://www.scmp.com/news/china/science/article/3320866/china-rocket-shortage-means-it-may-have-pick-favoured-candidate-take-starlink

the buildup of space debris. Advanced mission planning is essential for OOS systems to coordinate multiple service satellites, manage various tasks, and operate efficiently.

Based on this background, several key challenges emerge. These include designing low-cost and efficient constellation configurations for regional coverage, optimizing deployment strategies, and establishing sustainable mission planning for on-orbit services. Addressing these challenges is crucial for reducing constellation construction costs and promoting the commercial use of LEO technology. This research focuses on optimization problems throughout the lifecycle of LEO regional coverage constellations. By developing innovative models, solution methods, and simulations, this work aims to provide a technical framework for the complete lifecycle management of regional coverage constellations, supporting their design, deployment, operation, and long-term sustainability.

1.2 Introduction of LEO Constellations

The development of satellite constellations originated from aerospace exploration and military needs in the mid-20th century. Early satellite constellations were primarily used for positioning and navigation. In the 1960s, the first operational satellite constellation, known as the U.S. Navy's Transit navigation system, was deployed. This system consisted of 6 satellites in polar orbits (at approximately 1,100 km altitude) and provided navigation information for submarines. In the early 1990s, the United States deployed the well-known Global Positioning System (GPS) constellation [1], which adopted the classic Walker constellation configuration, with 24 satellites uniformly deployed in Medium Earth Orbit (MEO) at 20,180 km altitude (6 orbital planes with 4 satellites per plane). In the 21st century, China, Russia, and the European Union successively began deploying their own global navigation systems. The European Union's Galileo constellation [2] adopted a configuration similar to GPS, with 24 satellites uniformly distributed across 3 MEO orbital planes and 6 additional backup satellites. China's BeiDou-3 global navigation system [3], completed in 2020, also includes 30 satellites, with 24 satellites using the Walker configuration positioned in MEO at 21,500 km altitude. The remaining 6 satellites are deployed in Geostationary Orbit (GEO) and Inclined Geosynchronous Orbit (IGSO) to provide signal enhancement for China and the Asia-Pacific region. In recent years, with the rise of autonomous driving, mobile IoT, and other industries, demand for higher satellite navigation precision has increased. To address this, the industry has proposed establishing LEO navigation enhancement constellations. In 2017, China launched the world's first LEO navigation enhancement technology test satellite, which successfully completed performance testing and transitioned to operational service. However, as of now, no LEO navigation enhancement constellation has been built or is under construction globally.

Since the 1990s, the enormous potential of LEO satellite constellations for global communications has been recognized. One of the earliest LEO communication constellations was the Iridium constellation [4], led by Motorola. Deployed between 1997 and 2002, it comprised 66 satellites operating in polar orbits at 781 km altitude, using a Polar configuration to achieve continuous global coverage. The primary objective of the Iridium constellation was to provide low-throughput voice and data communication services to global users. However, due to high satellite manufacturing and launch costs at the time, coupled with insufficient market demand, Iridium's commercial operations faced difficulties, ultimately leading to bankruptcy protection in 1999. Nevertheless, the successful deployment of the Iridium constellation established a technical foundation for the development of LEO communication constellations. Concurrently, another significant LEO communication constellation was the Globalstar constellation [5], consisting of 48 satellites operating in inclined orbits at 1,414 km altitude, primarily addressing voice communication and low-speed data transmission needs. The Globalstar constellation began operations in 1998, but similar to Iridium, its commercialization process was hampered by high construction costs and limited market demand. It is evident that early LEO constellations were constrained by costs; although larger than mid-to-high orbit navigation constellations, they still maintained a scale of several dozen satellites and predominantly employed polar or inclined orbits, mainly for voice communication and low-speed data transmission.

As satellite manufacturing and launch costs continued to decrease, commercial companies began proposing concepts for massive LEO satellite internet constellations in the 2010s. These constellations typically include hundreds or even tens of thousands of LEO satellites to achieve dense, continuous coverage of the Earth's surface, supporting high-throughput internet data transmission for ground users. The most notable example is SpaceX's Starlink mega-constellation [6]. As of late 2024, Starlink has achieved operational status with 6,764 satellites in orbit. This constellation primarily uses the Walker configuration, with satellites mainly deployed in orbits at approximately 550 km altitude, providing all-weather, multiple coverage for latitudes within ±60°. It has been successfully applied to high-speed data communication in civil aviation, maritime transportation, and military applications. In the future, Starlink will expand to include coverage layers at 340 km and 1,150 km altitudes, ultimately forming a multi-layer mega-constellation with over 12,000 satellites in orbit [6]. The OneWeb constellation [7] concept predated Starlink, first proposed in 2014. This constellation consists of 648 satellites in a Polar configuration for global coverage and is currently nearing completion of its full satellite deployment with commercial operational capability. China also launched the construction of two major LEO internet constellations in 2024: Qianfan [8] and Guowang [4]. As of April 2025, approximately 90 satellites have been deployed for the Qianfan constellation. The constellation operator plans to achieve regional network coverage with 648 satellites by the end of 2025 and ultimately reach approximately 15,000 operational satellites by the end of 2030 [8].

LEO constellations are also used for remote sensing. Due to sensor range limitations, remote sensing constellations typically cannot achieve continuous ground coverage; their primary purpose is to enable rapid revisits of global or specific regions. A typical case is the PlanetScope small satellite constellation [9], which has undergone three iterations since 2014 and includes approximately 130 satellites operating in Sun-Synchronous Orbit (SSO), capable of imaging approximately 200 million square kilometers daily. China's Chang Guang Satellite Technology Co. has also been building the "Jilin-1" LEO remote sensing constellation since 2015, comprising 138 satellites that can achieve revisit times as short as 10 minutes for any global location [10]. To meet various ground observation resolution requirements, these constellations typically arrange satellites at different SSO altitudes without forming standard constellation configurations. However, due to the characteristics of SSO, the orbital planes between satellites remain relatively fixed.

Table 1.1 summarizes the representative constellations mentioned above for communication, navigation, and remote sensing, along with their main parameter configurations. It can be observed that to achieve better communication performance and imaging quality, the focus of constellation construction has shifted from GEO and MEO to LEO, while the number of satellites has continuously increased. However, the vast majority of LEO constellations aim to achieve global coverage, with satellite numbers typically ranging from

[4] https://spacenews.com/china-kicks-off-guowang-megaconstellation-with-long-march-5b-launch/
[5] https://en.wikipedia.org/wiki/Indian_Regional_Navigation_Satellite_System
[6] https://www.eoportal.org/satellite-missions/jilin-con#spacecraft

Table 1.1: Summary of typical satellite constellations.

Constellation	Application	Configuration	Coverage Range	Number of Satellites	Orbital Altitude (km)	Orbit Type	Construction Period (years)	Country
GPS [1]	Navigation	Walker	Global	24	20180	MEO	1989-1993	USA
Galileo [2]	Navigation	Walker	Global	30	23222	MEO	2011-present	EU
BeiDou-3 [3]	Navigation	Walker	Global	30	21500, 35786	MEO,GEO, IGSO	2015-2020	China
IRNSS [5]	Navigation	\	Regional	7	35786	GEO,IGSO	2013-2021	India
Iridium [4]	Communication	Polar	Global	66	781	LEO	1997-2002	USA
Starlink [6]	Communication	Walker	$\pm 60°$ latitude	12000	550, 340, 1150	LEO	2019-present	USA
OneWeb [7]	Communication	Polar	Global	648	1200	LEO	2019-present	UK
Qianfan [8]	Communication	Polar	Global	15000	1000	LEO	2024-present	China
PlanetScope [9]	Remote Sensing	\	Global	130	450 - 525	SSO	2014-2022	USA
Jilin-1 [6]	Remote Sensing	Coverage band	Global	138	500 - 650	SSO	2015-present	China

Data acquired by October 2025.

hundreds to tens of thousands. Due to their predominantly symmetric configurations, they may struggle to meet the complex coverage requirements of specific regions.

1.3 Introduction of Satellite Launch and Deployment Methods

Constellation launch and deployment (L&D) refers to the process of launching the satellites to space and placing them into their designated orbits. It is a key part in the construction process of LEO satellite constellations, with its efficiency and cost directly affecting the overall performance and commercialization process of the constellation. Currently, there are two main launch methods [11], [12]: (1) Direct inject, where the launch vehicle directly launches satellites into their operational orbits. This typically utilizes solid-fuel launch vehicles that can quickly implement launch missions, characterized by rapid response capabilities and useful for quickly replenishing failed satellites in a constellation. The disadvantage is the limited number of satellites that can be carried in a single launch.

(2) Indirect inject, where the launch vehicle delivers satellites to an injection orbit (also called a parking orbit), after which the satellites autonomously maneuver to their operational orbits. Taking the Starlink constellation's launch and deployment as an example, a Falcon rocket can carry over 20 satellites at once to an injection orbit, after which each Starlink satellite gradually raises to its predetermined orbital altitude using onboard ion thrusters [7]. The advantage of indirect inject is the ability to simultaneously deliver multiple satellites to the same injection orbit, after which satellites autonomously maneuver to their operational orbits using onboard propulsion systems, significantly reducing launch costs. In comparison, direct inject typically requires individual launches for each satellite, resulting in higher costs [11], [13].

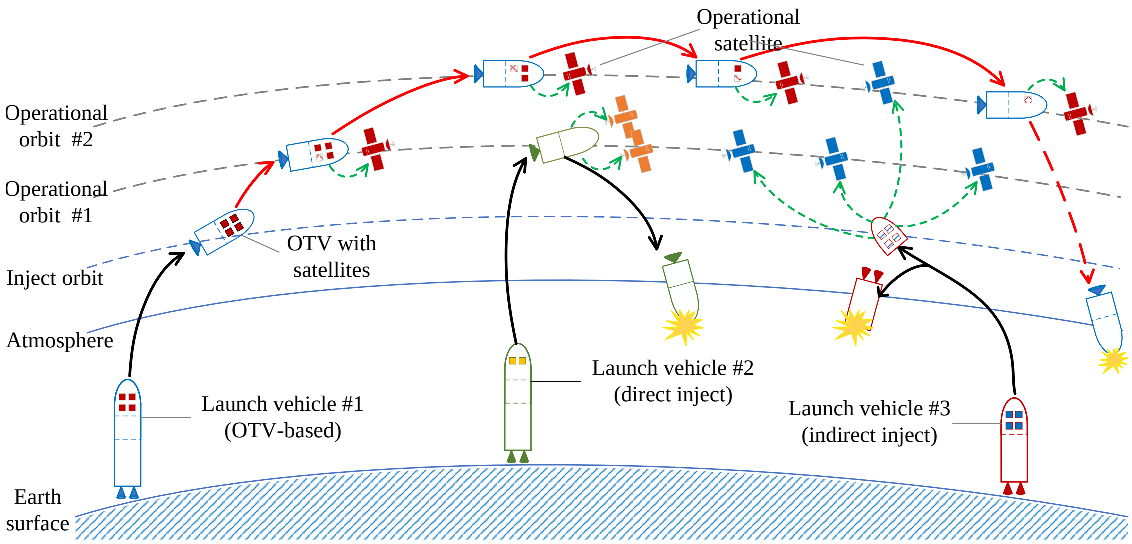


Figure 1.3: Schematic comparison of the three types of satellite launch methods. The black arc represents the launch and de-orbit process. The red arc represents the orbital maneuver process of the OTV. The red dashed arcs represent the de-orbit process of the OTV. The green dashed arc represents the satellite release and autonomous orbital maneuver process.

In recent years, as LEO satellite constellations have grown in scale and constellation configurations have become increasingly complex, the limitations of the above two meth-

[7] https://www.theverge.com/2019/5/15/18624630/

ods have gradually become apparent. The direct inject method offers fast response but at higher costs, making it difficult to meet the deployment needs of large-scale satellite constellations. The indirect inject method, while improving launch efficiency, still requires satellites to perform multiple maneuvers in the injection orbit, increasing satellite fuel consumption and complexity, and potentially leading to satellite orbital insertion failures.

To address these challenges, in recent years, some aerospace companies and research institutions have begun exploring new launch methods, such as using Orbital Transfer Vehicles (OTVs) for rapid satellite deployment [14]–[16]. An OTV is a spacecraft equipped with high-thrust and multiple restart capabilities, able to perform multiple orbital maneuvers in space to deploy satellites sequentially into different orbital positions, significantly enhancing launch efficiency and flexibility. In this mode, the launch vehicle first delivers the OTV and satellite combination to an injection orbit, after which the OTV carries the satellites through multiple orbital maneuvers, delivering them sequentially into their predetermined orbits. Figure 1.3 shows a process comparison of the three launch and deployment methods described above.

Multiple countries and institutions have successfully used OTVs for satellite launch and deployment. Typical examples include China's "Yuanzheng" series upper stage, with the "YZ-2" upper stage successfully used for deploying the 01 and 03 groups of satellites in the "Guowang" mega-constellation [8]. The ION satellite carrier (Figure 1.4) developed by Italy's D-Orbit company has successfully deployed over one hundred LEO satellites [9]. Meanwhile, many institutions and companies are actively developing OTV technology, such as France's Hybrogines Company's Hyscab, Exotrail Company's spacevan (Figure 1.5) and the European Space Agency's ASTRIS, and so on [10].

[8] https://calt.spacechina.com/n482/n498/index.html
[9] https://www.dorbit.space/launch-deployment
[10] https://europeanspaceflight.com/european-otvs-index/

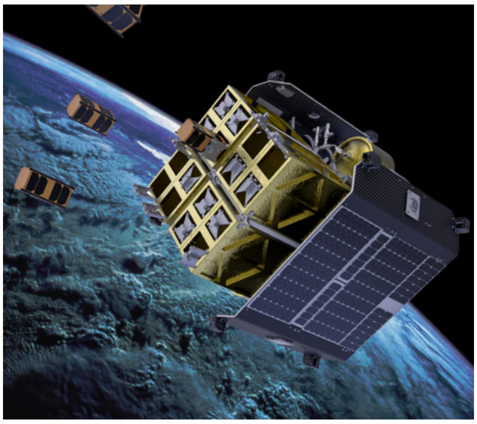


Figure 1.4: ION satellite carrier. [11]

Figure 1.5: Exotrail's spacevan. [12]

Compared to direct and indirect inject methods, the OTV-based launch and deployment method not only extends satellite lifespans by avoiding consumption of satellite fuel [17] but also accelerates the deployment process of satellite groups through powerful propulsion systems. Overall, the application of OTVs makes the satellite constellation launch and deployment process more efficient and flexible, with this launch method showing significant potential in the construction of LEO satellite constellations with frequent requirements.

1.4 Introduction of Satellite On-Orbit Servicing Methods

Currently, LEO constellations are still in the large-scale deployment and construction phase, with relatively little consideration given to satellite maintenance, updates, and replacements during the long-term operation after constellation completion. However, as the scale of LEO constellations continues to expand and their operational time increases, there will inevitably be satellites in the constellation that fail or exhaust their fuel and

[11] https://www.dorbit.space/ion-satellite-carrier
[12] https://www.exotrail.com/in-orbit-services

can no longer function normally. Such satellites, having lost their autonomous control capabilities, not only occupy valuable orbital resources, leading to degraded constellation performance, but also pose collision risks with other satellites, resulting in space environment pollution. To achieve efficient, low-cost disposal of such satellites, OOS technology is an promising solution.

Satellite on-orbit servicing refers to various service operations performed on spacecraft in orbit, including maintenance, repair, upgrades, or life extension. Its core objective is to extend spacecraft life, enhance functionality, or resolve faults through active intervention, thereby reducing space mission costs, decreasing space debris, and promoting the sustainable use of space resources. Early practices of the OOS concept included multiple on-orbit repairs and upgrades to the Hubble Space Telescope, as well as regular resupply missions to space stations. However, these on-orbit service operations primarily relied on human operations, resulting in high mission costs. To reduce the cost of future on-orbit services, since this century, the development direction of related technologies has been to achieve autonomous task execution by service satellites, completing complex operations such as target capture, component replacement, and surface inspection through end effectors like robotic arms.

Currently, multiple countries have successfully conducted on-orbit verification of autonomous satellite servicing operations, encompassing various types such as on-orbit refueling, orbital correction, and maintenance. A notable example is the Mission Extension Vehicle (MEV) project led by Northrop Grumman Corporation in the United States, designed to provide orbital maintenance and refueling services for GEO satellites that have depleted their fuel. This project successfully completed the launch and on-orbit mission verification of MEV-1 and MEV-2 spacecraft in 2019 and 2020, respectively [18]. The company also plans to conduct a second mission in 2026, aiming to launch two kinds of spacecrafts of Mission Extension Pod (MEP) modules and Mission Robotic Vehicle (MRV) spacecraft to cooperatively provide more diversified on-orbit life extension and repair services for multiple client satellites [19]. Additionally, China's Tianyuan-1 satellite successfully conducted an on-orbit refueling experiment in 2016, establishing a foundation for future OOS missions. Another relatively mature category of OOS missions is space debris removal. China's "Aolong-1" satellite successfully conducted a space debris capture experiment in 2016, marking further development of China's OOS technology [20]. Furthermore, the RemoveDEBRIS project from the University of Surrey in the UK also successfully conducted space debris capture and removal experiments [13]. China's "Shijian-21" satellite [21], launched in 2021, successfully captured the decommissioned BeiDou-2 G2 satellite in GEO and towed it to a graveyard orbit, marking a technological breakthrough in large space debris capture and removal.

However, more complex OOS technologies, such as on-orbit repair and assembly, which often require collaboration among service satellites, remain largely in the conceptual research stage and have not yet undergone large-scale on-orbit testing [22]. The few on-orbit service tests that have been conducted include Japan's Engineering Test Satellite No. 7 (ETS-VII) mission in 1997 [23]. Similarly, the United States' Orbital Express program only achieved the replacement of a fixed-shape battery on a target satellite, demonstrating limited versatility [24].

As geopolitical conflicts intensify, on-orbit inspection technology for examining the operational status of unknown satellites has gained increasing attention from various countries. The United States' Geosynchronous Space Situational Awareness Program (GSSAP) was also successfully implemented in 2016 [25], aiming to provide continuous monitoring and tracking of GEO space targets. The "USA-271" satellite from this program

[13] https://spacenews.com/removedebris-mission-confirms-launch-in-2017-using-the-iss/

conducted multiple orbital maneuvers in July 2021 to secretly approach and reconnoiter China's "Shijian-20" satellite, posing a significant security threat to the Chinese satellite [14].

Table 1.2: Summary of typical OOS missions.

Service Type	Typical Mission	Year	Country
Orbital Correction and Maintenance	MEV-1/2 [18]	2019	USA
	MEP [19]	2026(planned)	USA
Refueling and Supply Replenishment	Tianyuan-1 [15]	2016	China
	Tanker-001 [26]	2021	USA
On-Orbit Repair	ETS-VII [23]	1997	Japan
	Orbital Express [24]	2007	USA
	MRV [19]	2026(planned)	USA
Space Debris Removal	Aolong-1 [20]	2016	China
	RemoveDEBRIS [16]	2018	UK
	Shijian-21 [21]	2021	China
On-Orbit Inspection	BanXing-1 [17]	2008	China
	GSSAP [25]	2016	USA

Table 1.2 summarizes typical OOS missions that various countries have implemented or plan to implement, categorized by different types of on-orbit services. It provides the definitions of different on-orbit service tasks, typical examples, and their implementation years and countries. Overall, the development of OOS technology is trending from human participation to autonomous intelligence, from single service satellites to multi-satellite coordination, and from government-led initiatives to commercial market-oriented approaches.

[14] https://www.scmp.com/news/china/military/article/3154496/chinese-satellite-shows-space-warfare-prowess-dodging-us

[15] https://www.chinadaily.com.cn/china/2016-07/04/content_25954551.htm

[16] https://spacenews.com/removedebris-mission-confirms-launch-in-2017-using-the-iss/

[17] https://space.skyrocket.de/doc_sdat/banxing.htm

1.5 Research Aims and Significance

This dissertation focuses on system optimization problems throughout the lifecycle of LEO regional coverage constellations, which primarily consists of three key phases: constellation design, launch and deployment, and OOS. Following this lifecycle logic, this research addresses three core challenges. First, it aims to develop an efficient design methodology for regional coverage constellations to minimize costs while meeting specific coverage requirements. Second, it investigates optimization methods for constellation L&D under new operational modes, balancing multiple objectives and complex constraints. Third, it explores optimal mission planning strategies for OOS to ensure the long-term sustainability of the constellation. By addressing these interconnected problems, this research seeks to establish a comprehensive technical framework for the complete lifecycle management of regional coverage constellations, providing technical support for their design, deployment, and sustainable operation through innovative modeling, advanced solution techniques, and simulation.

At the theoretical level, this dissertation addresses the limitations of existing research in the lifecycle management of LEO constellations. Current constellation design methods often focus on global coverage and lack efficient, low-cost solutions for specific regional demands. Similarly, launch and deployment strategies have not fully adapted to the complexities of new paradigms, such as OTV-based deployment, often failing to balance conflicting objectives like cost, time, and fuel consumption. Furthermore, mission planning for on-orbit servicing is still in its early stages, necessitating integrated frameworks capable of coordinating multiple service assets. To bridge these gaps, this dissertation integrates orbital dynamics, intelligent optimization algorithms, and operations research to construct a multi-stage optimization framework for the "design-deployment-on-orbit servicing" lifecycle of LEO constellations. This work aims to enrich the theoretical foundation in the field of aerospace systems engineering.

At the application level, the dissertation tries to provide valuable technical reference for the construction of new generation Internet satellite constellations. Additionally, the proposed launch and deployment optimization methods can significantly help aerospace enterprises reduce constellation deployment costs and improve satellite deployment efficiency. Furthermore, the on-orbit refueling mission planning approach for LEO constellations contributes to the efficient operation of OOS systems, alleviating the scarcity of LEO space resources and preventing further accumulation of space debris, thereby promoting sustainable development of the space environment.

1.6 Thesis Structure

Building on recent research advances and identifying current gaps, this work systematically explores regional coverage constellation design, launch deployment optimization, and on-orbit servicing mission planning. The main research content and chapter structure are as follows:

Chapter 1: Introduction. This chapter introduces the research background and significance, systematically analyzes the current status of LEO satellite constellations, launch and deployment technologies, and on-orbit servicing technologies. Finally the research objectives and content of this thesis are outlined.

Chapter 2: Literature Review. This chapter reviews relevant studies on LEO constellation design, launch and deployment, and on-orbit servicing. It also analyzes the limitations of existing research and lays the foundation for subsequent chapters.

Chapter 3: Regional Coverage LEO Constellation Design Based on Repeat Ground

Track (RGT) Orbits. This chapter first introduces a satellite–ground target coverage analysis method based on circular convolution, establishing a relationship between constellation pattern vectors and target visibility vectors for efficient modeling of regional coverage performance. The single RGT orbit constellation design model is then extended to a multi-RGT orbit synchronous orbit optimization and satellite deployment (SOOSD) model for coordinated optimization of orbital parameters and satellite deployment locations. To address the model's high-dimensional, sparse, and nonlinear constraints, a two-stage optimization framework integrating differential evolution algorithms and integer programming solvers is designed. Multiple simulation cases with varying geographic distributions and coverage requirements demonstrate the proposed method's significant advantages in satellite count, coverage performance, and computational efficiency.

Chapter 4: Bi-objective Optimization of LEO Constellation L&D Based on OTVs. This chapter describes the OTV-based LEO constellation launch deployment scenario and derives a rapid calculation method for OTV fuel consumption. A bi-objective combinatorial optimization model is established to simultaneously optimize launch combinations and deployment sequences, minimizing launch frequency and average fuel consumption. Then a customized solution framework integrating the ϵ-constraint method with cluster and nearest neighbor sorting initialization and local search operators in a genetic algorithm (CNS-LSGA) is proposed. Comprehensive simulations and comparative experiments on repeat ground track-based regional coverage constellations and multi-layer Walker constellations validate the method's superiority in solution quality, robustness, and diversity.

Chapter 5: Mission Planning for LEO Constellation On-orbit Servicing Systems. This chapter addresses the mission planning problem for LEO constellation OOS systems, proposing a two-stage optimization framework for the "service station-servicing satellite" OOS mode. First, an modified spectral clustering and nonlinear programming (MSC-

NLP) method is proposed to achieve coordinated optimization of client satellite assignment and optimal orbital parameters for multiple service stations. Second, for servicing satellite task scheduling, a mixed-integer nonlinear programming model is established, and a quantum genetic algorithm (GQA) is designed for efficient solution, enabling simultaneous optimization of fuel consumption and servicing satellite utilization. Comparative studies with various optimization algorithms in both small and large-scale simulation scenarios demonstrate the proposed method's advantages in fuel consumption and servicing satellite utilization, as well as its scalability across different scenario sizes.

Chapter 6: Conclusion. This chapter summarizes the main research contribution and innovations of the thesis and provides an outlook on future research directions.

Chapter 2

Literature Review

2.1 Literature Review of LEO Constellation Design

The essence of LEO constellation design lies in how to optimally configure the number and orbital positions of satellites under specific resource constraints to achieve coverage requirements for designated ground targets while optimizing given design objectives such as maximizing quality of service [27], minimizing satellite usage [28], minimizing target revisit time [29], and so on.

Coverage performance is a key metric in constellation design. Constellation designs are classified based on coverage area, duration, and multiplicity. Coverage area can be global, latitudinal, regional, or point-specific [28], [30]–[32]. Coverage duration can be continuous or discontinuous [33]–[36]. Coverage multiplicity refers to the number of satellites simultaneously covering a target, which can range from single to multiple-fold [30]. Methods for continuous global or latitudinal coverage date back to the 1960s [37], leading to well-known patterns like Walker, Street-of-Coverage, and Rosette configurations [27], [34], [38]–[40]. These designs typically use symmetrical structures. Later, genetic algorithms were used to explore asymmetric configurations, which showed improved performance over traditional Walker patterns [41]–[43]. While effective for medium and high orbit constellations, these methods are less practical for LEO constellations. The limited coverage of a single LEO satellite means that achieving global coverage requires hundreds or thousands of satellites. This high number significantly increases costs, making such constellations affordable for only a few nations with advanced space programs. Moreover, large, symmetrical LEO constellations face other issues: (1) their uniform structure struggles to meet specific regional demands; (2) global connectivity raises security and data privacy concerns [44]; and (3) the large number of satellites causes light pollution and electromagnetic interference, which can disrupt astronomical observations [45].

In contrast, regional coverage LEO constellations typically require only a small number of satellites to meet specific regional requirements such as rapid revisit times or other complex coverage needs. The significant reduction in satellite quantity substantially lowers construction and launch costs, making these constellations well-suited for smaller nations seeking communications and remote sensing capabilities within their territories, thus lowering the barriers to entry for LEO constellation development. Additionally, regional coverage constellations can be deployed as components of global coverage constellations, enabling quick revenue generation when market demands are not yet fully defined, demonstrating greater flexibility and scalability [46].

Given these advantages, regional coverage constellation design has gained increasing attention. Existing regional coverage constellation designs do not need to meet continuous coverage requirements but instead focus on achieving rapid revisit times or other complex coverage requirements in specific regions [47]–[49]. Ulybyshev [35], [50] proposed a constellation visibility analysis method based on the (Ω, u) (right ascension of ascending node-argument of latitude) phase plane and applied it to regional coverage constellation design to minimize revisit time for regional targets or latitude intervals. In regional coverage constellations, orbital parameters for each satellite can be independently selected based on coverage requirements for the target region. Compared to fixed parameter constellations like Walker, this approach provides more optimization variables and better continuity in the decision space. Consequently, researchers have successfully employed intelligent optimization algorithms for regional LEO constellation design. Mailhot [51] used genetic algorithms for LEO regional navigation constellation design, demonstrating that genetic algorithm results required fewer satellites than enumeration methods using 3D lattice Flower or Walker constellation patterns. Wang et al. [52] proposed an improved particle swarm algorithm for constellation design considering complex sensor fields of view to maximize regional target coverage.

A promising approach for more efficient regional coverage is the implementation of Repeat Ground Track (RGT) orbits. Satellites following this orbital pattern can revisit the same ground area within a fixed period. This characteristic can satisfy complex coverage requirements for regional targets [53], [54]. Constellation design problems based on RGT orbits have been extensively studied. He et al. [55] investigated global discontinuous coverage constellations using RGT orbits , with simulation results showing that hybrid RGT constellations outperform Walker constellations in coverage performance. Chadalavada et al. [48] designed an RGT orbit-based asymmetric constellation by a satellite coverage map-based optimization framework for regional coverage in hurricane monitoring applications. Dong et al. [47] developed a novel revisit time image extrapolation method to accelerate coverage analysis for RGT orbits and designed a lookup table-based optimization method for discontinuous target coverage. Lee [29] developed an optimization method for satellite constellations targeting multiple point objectives, considering observation priorities and weights for dispersed targets.

However, these studies primarily focus on discontinuous coverage for Earth observation applications by minimizing revisit time, making them less suitable for regional communication systems that require continuous or complex temporal coverage. To bridge this gap, Lee et al. [28] formulated the constellation design problem as a linear integer programming model by leveraging the circular convolution property between constellation pattern and target visibility vectors. This approach allows complex spatiotemporal coverage requirements to be integrated as linear constraints. A key limitation, however, is that the model only optimizes satellite positions on pre-selected RGT orbits without optimizing the orbital parameters themselves, thus restricting the search space and potentially yielding only locally optimal solutions. In a subsequent study, Lee et al. [56] proposed a general constellation reconfiguration model applicable to non-RGT orbits, solved with an efficient Lagrangian relaxation-based heuristic. Nevertheless, this model also did not optimize the candidate orbit parameters. Later, Soung Sub Lee [57] introduced a method that co-optimizes RGT orbit parameters and satellite distribution, enabling a more thorough exploration of the solution space. However, this method is only applicable to scenarios with discontinuous target coverage and a small number of satellites.

2.2 Literature Review of Constellation Launch and Deployment Optimization

Constellation launch and deployment (L&D) optimization involves making rational decisions about launch timing, launch sites, satellite combinations, and orbital maneuvers under given constellation orbital positions and parameters of satellites and launch vehicles, with the aim of achieving cost-effective, efficient, and reliable constellation deployment.

(1) Launch phase optimization

With the continuous expansion of LEO satellite constellations, research on constellation L&D optimization has garnered increasing attention in recent years. Current research mainly focus on the direct and indirect inject mode, with relatively limited research on OTV-based launch methods. Colombi et al. [58] investigated heterogeneous satellite constellation L&D optimization, considering both satellite-launch vehicle allocation and the selection of launch sites and orbits. This problem was modeled as an integer programming problem and validated through case studies of meteorological and navigation satellite constellations. Sung et al. [11] studied a similar problem and designed a two-stage optimization framework to solve the L&D optimization problem, additionally considering the choice between direct inject and indirect inject methods, though without detailed discussion of the satellite deployment process. Di Pasquale et al. [59] established a hybrid integer multi-objective optimization model based on the indirect inject method using multidisciplinary optimization techniques, which simultaneously optimizes launch timing, injection orbits, and orbital maneuvers, verifying the effectiveness of this method through numerical simulations. Barato et al. [13] conducted the first comparative study of the comprehensive efficiency of the three launch methods mentioned above, finding that the indirect inject method was the most cost-effective, but the Orbital Transfer Vehicle (OTV)-based method was the overall optimal approach when satellite orbital maneuvering capability was limited. The authors also provided the most suitable L&D methods for different launch scenarios.

(2) Deployment phase optimization

The research on L&D optimization problems mentioned above mostly focuses on the launch phase, therefore, decision variables are mostly concentrated in the launch window, launch site selection, etc.However, the decision space for those decision variables are typically limited, while factors affecting launch efficiency are concentrated in the deployment process, especially in indirect inject and OTV-based launch modes. Consequently, some scholars have focused on studying orbital maneuver strategies during constellation deployment process. Bakhtiari et al. [60] proposed a multi-satellite autonomous orbital maneuvering strategy based on multi-pulse Lambert orbital transfer methods for multi-plane satellite constellation deployment, validating its effectiveness through numerical simulations. Leppinen [61] introduced a novel strategy utilizing atmospheric drag for multi-plane deployment of constellation satellites, which significantly reduces satellite fuel consumption compared to traditional orbital maneuver methods, thereby extending satellite orbital lifetime, although the maneuvers require a longer time. Subsequently, Di Pasquale et al. [62] employed a maneuvering strategy combining J_2 perturbation effects and finite thrust to balance fuel requirements and deployment time in multi-plane constellation deployment, establishing a multi-objective optimization model validated through the FORMOSAT-3/COSMIC mission. Liu et al. [63] designed a Kuhn-Munkres matching deployment optimization method based on the indirect launch method, optimizing the process of deploying satellites from different initial states to different phases of the same orbit, significantly reducing satellite fuel consumption during deployment. Xue et al. [64] optimized the injection orbit position of launch vehicles by considering J_2 perturbation effects and finite thrust in the satellite deployment process. Roychowdhury et al. [65] studied the deployment of NanoFF satellite clusters using ION satellite carriers, providing deployment strategies that maximize collision probability reduction while considering multiple practical mission constraints, analyzing deployment position accuracy, and offering recommendations for improving deployment strategies.

(3) L&D optimization for OTV-based launch methods

Although the above studies have explored deployment strategies utilizing atmospheric drag or J_2 perturbation effects for orbital transfers, these methods differ fundamentally from OTV-based L&D methods in orbital transfer mechanisms and optimization requirements. OTV-based L&D strategies introduce unique challenges due to their reliance on high-precision multi-objective orbital maneuvers that consume significant amounts of fuel. Therefore, the deployment sequence and orbital positions of satellites are crucial for total fuel consumption, determining the feasibility and success rate of each launch mission, thereby further affecting the overall cost of launching and deploying the entire constellation. The OTV-based L&D optimization problem can be viewed as a Multi-Spacecraft Successive Rendezvous Planning problem (MSSRP) [66], [67]. Such problems are typically modeled as dynamic versions of the Traveling Salesman Problem (Time Dependent Travel Salesman Problem, TDTSP) [68], [69] or Vehicle Routing Problem (VRP) [70]. Since decision variables involve both continuous and discrete variables, and it is difficult to obtain derivatives of objective functions, current methods for solving multi-objective continuous orbital maneuver problems mainly employ meta-heuristic algorithms, such as Genetic Algorithm (GA) [68], Particle Swarm Optimization (PSO) [70], and Simulated Annealing (SA) [17]. These algorithms perform well in scenarios with low orbital transfer velocity increments (e.g., satellites in GEO or IGSO) and when extended transfer times can be leveraged to utilize J_2 perturbation effects. However, OTV-based L&D processes require deploying satellites to multiple orbital positions (even across orbital planes) in a short period, significantly increasing fuel consumption. In such cases, the limited fuel of the OTV restricts the feasible solution space, causing meta-heuristic algorithms widely adopted in existing literature to easily fall into local optima or even fail to find feasible solutions [71], [72]. Therefore, how to achieve efficient mission optimization for OTV-based L&D processes remains a problem requiring further research.

2.3 Literature Review of On-orbit Servicing Mission Planning

2.3.1 Classification of OOS Mission Planning Problems

From the perspective of mission types, different missions impose different terminal constraints on rendezvous. These can be classified into four categories: (1) On-orbit repair [73], [74]: These missions require precise rendezvous between service satellites and CSs, necessitating equal terminal position and velocity. While the service satellite's mass doesn't change significantly before and after the mission, mission planning models must account for varying repair service times for different satellites. (2) On-orbit inspection [75], [76]: These missions require service satellites to maintain relative position and velocity within certain ranges during rendezvous, which increases the decision space and variables for terminal positions, thereby increasing solution complexity. (3) On-orbit refueling (OOR)/component replacement [17], [68], [77]: After rendezvous, the service satellite transfers fuel/components to the CS, causing significant mass changes to the service satellite. These mass changes affect fuel consumption for subsequent missions, creating sequence-dependent transfer costs between targets and increasing problem complexity. (4) Debris removal [78]–[80]: This typically involves either attaching small deorbiting devices to debris (similar to refueling missions) or capturing debris and transporting it to a graveyard orbit, which requires consideration of orbital and fuel changes before and after servicing. Mission types are usually determined by practical requirements, with different types imposing different terminal constraints on orbital maneuvers, affecting modeling approaches and solution methods. Some research also considers mixed-type missions, further increasing planning complexity [81], [82].

According to the service satellite deployment methods and quantities, OOS missions can be categorized as follows: (1) “One-to-many” mode [68], [83], [84]: A single service satellite is launched to a predetermined orbit to perform multiple service missions sequentially. (2) “Many-to-many” mode [78], [85]: Multiple service satellites are deployed to specific orbits, each executing pre-assigned service tasks sequentially. (3) “Peer-to-Peer (P2P)” mode [86]–[88]: No distinction between service satellite and CSs; all spacecraft can provide and receive services. (4) “Service station-service satellite” mode [70], [89], [90]: A large on-orbit service station (also called ”fuel depot” or ”space station” in existing research) is pre-deployed in space. And multiple service satellites depart from the SS with necessary payloads and fuel to perform on-orbit services for multiple CS. (5) “Service satellite-function package” mode [91]: Function packages of different types are pre-positioned near CSs; service satellites first maneuver to capture these packages, then rendezvous with CSs to install the packages. Given current space technology, ”one-to-many” and ”many-to-many” modes are relatively easy to implement but typically only capable of performing specific tasks in the short term. The P2P mode requires all spacecraft to carry complete on-orbit service payloads, making it difficult to meet diverse service needs and impractical for real applications. Simulation studies in [92] show that P2P mode outperforms ”many-to-many” deployment strategies only when targets meet certain conditions. The ”service station-service satellite” mode imposes fewer requirements on service satellites, providing high flexibility through service stations carrying various necessary supplies. With regular resupply missions to and from the service station, this mode enables long-term servicing of space objects, offering greater economic benefits [93]. The ”service satellite-function package” mode was proposed by Northrop Grumman’s MEP+MRV project but remains in the conceptual stage without on-orbit validation and lacks research on mission planning methods, raising questions about its effectiveness.

2.3.2 "One-to-many" and "Many-to-many" OOS Mission Planning Problems

The mission planning problems in "one-to-many" and "many-to-many" modes can be viewed as multi-target continuous orbital rendezvous problems [94], with different types of missions imposing different terminal rendezvous constraints. These problems are typically modeled as Mixed Integer Nonlinear Programming (MINLP) problems, where decision variables include the rendezvous sequence of CSs and orbital maneuver timing. Due to the complexity of having both continuous and discrete variables, most existing literature employs meta-heuristic algorithms for solutions.

For OOR mission, Zhang et al. [68] proposed a hybrid-encoded GA to determine the refueling sequence, service timing, and orbital transfer timing in "one-to-many" OOR mission scheduling problems. Du et al. [95] proposed a cooperative orbital maneuver strategy for target and service satellites in "many-to-many" modes, with simulations showing this strategy significantly conserves fuel compared to traditional OOR methods. Zhou et al. [96] modeled the "many-to-many" OOR mission planning problem for GEO satellites as a multi-objective optimization problem to simultaneously optimize mission completion time and service satellite fuel consumption, designing a two-stage solution framework. Sorenson et al. [97] modeled "many-to-many" OOR mission planning as a mixed integer linear model, studying OOR missions in LEO, MEO, and GEO orbits. In their case studies, they also investigated how initial service satellite positions affect mission completion rates.

Space debris removal mission planning is a widely studied OOS mission in "one-to-many" and "many-to-many" modes. Shen et al. [78] studied debris removal problems under J_2 perturbation, modeling the problem based on the Traveling Salesman Problem

(TSP) and using ant colony algorithms for solutions. Li and Baoyin [98] combined evolutionary algorithm mechanisms with ant colony algorithms to design an Evolving Elitist Club Algorithm (EECA) for solving debris removal mission planning problems, achieving better results than ant colony algorithms. Building on this, Zhang et al. [80], [99] proposed a Timeline Club Optimization (TCO) based on the basic Ant Colony Optimization (ACO) to solve Time-Dependent TSP-based debris removal mission planning problems. Simulation results showed that TCO outperforms traditional algorithms (GA, ACO, tree search). Bang et al. [100] designed a two-stage optimization framework for debris removal mission planning, using column generation algorithms at the upper level to determine removal sequences and trajectory optimization at the lower level. Chen et al. [101], based on the 9th Global Trajectory Optimization Competition (GTOC-9) [102], proposed using clustering methods for target allocation, achieving better solutions than the competition's winning team through multi-level optimization. Guo et al. [103] proposed a GA with multi-parameter serial encoding for "one-to-many" debris removal problems where service satellites carry multiple debris pieces to graveyard orbits. Zhu et al. [104] designed a three-stage optimization method of "removal sequence-rendezvous timing-transfer trajectory" for debris removal mission planning involving installation of deorbit function packages, converting the original continuous-discrete problem into a discrete optimization problem by discretizing transfer candidate time periods, achieving state-of-the-art results in GTOC9 cases.

Research on on-orbit inspection mission planning is also present in the literature. For instance, Zhou et al. [75] investigated mission planning for multiple LEO satellites inspecting GEO targets, proposing a two-stage optimization algorithm to minimize both fuel consumption and inspection time. Wu et al. [105] developed a hierarchical optimization framework for "many-to-many" inspection missions. Their approach utilizes the Consensus-Based Bundle Algorithm (CBBA) [106] for high-level task allocation and employs deep neural networks for low-level fuel optimization during orbital maneuvers.

Fewer studies have addressed mixed-mode OOS mission planning. Han et al. [81] studied multi-mode servicing for multiple GEO satellites using continuous low-thrust transfers. Sarton et al. [82] introduced a rolling horizon optimization strategy for OOS missions with diverse requirements and uncertain target needs. Daneshjou et al. [107] examined multi-type OOS mission in a "one-to-many" mode, which included inspection, debris removal, and refueling operations, and solved the problem using an improved PSO.

2.3.3 "Service Station-Service Satellite" OOS Mission Planning Problems

The "service station-service satellite" mode for OOS is commonly used for OOR mission. Mission planning in this mode typically divides into two parts: optimization of the service station's orbital position and mission planning for service satellites. The service station's orbital position optimization primarily aims to determine the optimal deployment location that enables service satellites to shuttle between the SS and CSs with minimal cost. Mission planning for service satellites involves determining the orbital maneuver rendezvous sequence and timing during mission execution to effectively service designated CSs.

Zhang et al. [89] first studied the optimal orbital design problem for GEO service stations by first setting candidate positions for the service station, then using an improved ant colony optimization algorithm to select the optimal service station orbit from these candidates. Results showed that the optimal service station orbit typically lies at the center of densely distributed CS clusters. Inspired by this, Zhu et al. [108] used an improved K-means method to cluster GEO CSs based on their orbital distribution and positioned service stations at the center of each cluster. Building on the effectiveness of clustering methods, Zhu et al. [90] further investigated the deployment of service stations for CSs in Sun-Synchronous Orbit (SSO). This research utilized the effects of Earth's non-spherical perturbation on low-orbit precession, implementing the Hybrid Density-Based Spatial Clustering Algorithm (HDBSCAN) to cluster CSs, and then using a GA-based Fuzzy C-means Clustering (GA-FCM) algorithm to optimize the service station orbital positions, achieving favorable results. Meng et al. [93] modeled the service station orbital deployment problem as a multi-objective optimization problem, determining orbital parameters by minimizing refueling response time and economic costs.

In the "service station-service satellite" mode, the mission planning problem for service satellites resembles that of "one-to-many" or "many-to-many" modes, with the difference being the need to consider the service satellite's return orbital maneuvers. Additionally, since service stations provide sufficient fuel, considerations sometimes include multiple round trips by service satellites for refueling, or even complex operations such as capturing CSs and returning them to the service station for redeployment. Zhou et al. [77] pioneered this problem by designing a two-layer optimization framework based on improved PSO to determine the refueling sequence and task execution time for each service satellite servicing GEO satellites, considering mid-mission returns for refueling, and achieving good results. Yan et al. [74] addressed on-orbit repair problems for GEO satellites, simultaneously considering cooperative repair by multiple service satellites and complex operations such as towing CSs back to facilities for repair. They designed a two-stage optimization framework using GA and Variable Neighborhood Search (VNS) algorithms to solve the mission planning problem, obtaining service sequences, repair strategies, and orbital maneuver timing for each service satellite. However, these studies predefined the service station's orbital position without jointly optimizing it with the service satellites' mission execution sequences. To address this, Zhu et al. [90] combined both aspects by obtaining a feasible solution for service satellite sequences and timing using the Gantt Chart Method (GCM) after determining the fuel station's orbital design. However, this method only aimed to obtain feasible solutions that satisfy constraints without incorporating optimization mechanisms, which may lead to greater fuel consumption and service satellite usage when handling large-scale problems.

2.4 Summary of Research Gaps

Existing research has achieved notable progress in LEO constellation design, launch and deployment optimization, and on-orbit servicing, several significant gaps remain. These can be summarized as follows:

First, regarding LEO constellation design, although global coverage constellation design methodologies are relatively mature, regional coverage constellation design approaches still require further development. Current research on regional coverage constellation design faces several challenges: (1) Most studies focus on discontinuous coverage of regional targets, with insufficient research on continuous coverage requirements for regional communication applications; (2) Some RGT-based design methods fail to consider the collaborative optimization of orbital parameters and satellite position distribution, resulting in limited search space that yields only locally optimal solutions; (3) There is limited research on constellation design methods addressing complex spatiotemporal coverage requirements for regional targets. Therefore, further exploration is needed to fully leverage RGT orbit characteristics for specific regional continuous or complex coverage requirements, achieving collaborative optimization of orbital parameters and satellite distribution.

Second, regarding LEO constellation L&D optimization, current research primarily focuses on indirect injection or a mix of direct and indirect injection methods. There is limited research on methods that use OTVs. The existing studies on OTV-based L&D optimization still have several shortcomings: (1) The number of satellites to be maneuvered is often predetermined instead of being an optimization variable. (2) The target orbits are typically similar, and the fuel constraints are not strict, making these methods unsuitable for deploying satellites into diverse orbits. (3) The solution methods are mostly general meta-heuristic algorithms, which are not tailored to the specific problem structure and are thus prone to finding local optima. Therefore, developing efficient mission optimization methods for OTV-based L&D that balance launch frequency and fuel consumption under complex resource constraints remains a significant challenge.

Finally, regarding satellite OOS mission planning, existing research primarily builds on “one-to-many” or “many-to-many” service modes, while research on “service station-service satellite” mode mission planning remains in its preliminary stages with limited contributions. Key limitations include: (1) Insufficient consideration of collaborative optimization between service station orbital positions and service satellite sequencing and timing, leading to limited system efficiency; (2) Inadequate systematic research on initial service station orbital deployment, with limited deployment scenarios considered, making it difficult to adapt to complex CS distributions; (3) Lack of systematic research addressing LEO satellite constellations with their large number of satellites and complex on-orbit servicing requirements. Therefore, further research is needed on “service station-service satellite” mode OOS mission planning methods for LEO constellations, achieving collaborative optimization of service station orbital positions and service satellite mission scheduling to meet future complex on-orbit servicing requirements.

Chapter 3

RGT Orbit-based Regional Coverage LEO Constellation Design

3.1 Introduction

Regional coverage LEO constellations have attracted considerable attention from both academia and industry due to their high flexibility, low launch and manufacturing costs, and enhanced communication security. Unlike global coverage constellations, regional constellations allow for the flexible configuration of satellite numbers and orbital parameters tailored to the specific demands of a target area, thereby significantly reducing system development and operational costs. However, prevailing constellation design methodologies predominantly focus on uniform global or zonal coverage, which often fails to meet the complex and varied coverage requirements of specific regions. Existing studies on constellation design for regional targets have largely concentrated on non-continuous coverage, emphasizing the minimization of revisit times. Consequently, achieving efficient continuous coverage for regional targets while minimizing the required number of satellites, especially under the constraints of limited orbital resources and complex coverage demands, remains a critical and unresolved challenge in the field.

To address this, Lee et. al.[28], [56] first proposed a design method for regional LEO communication constellations based on RGT orbits, which can theoretically satisfy the complex spatio-temporal coverage requirements of regional targets. This method formulates the constellation design as an integer linear programming problem. By leveraging the cyclic convolution property between the constellation pattern and the target's visibility timeline, it minimizes the number of satellites while ensuring coverage requirements are met. The model is applicable to both point and area targets, with pre-determined RGT orbit parameters. Nevertheless, this approach has several limitations: (1) The RGT orbit parameters are pre-defined, which may lead to low coverage efficiency for each satellite, thus requiring more satellites to meet the coverage demand. In practice, orbital parameters should be optimized based on the location of the target area. (2) For scenarios with small satellite coverage footprints (determined by the minimum elevation angle ε and the satellite's semi-major axis a) and widely distributed regional targets, a single RGT orbit may fail to provide effective coverage for all targets, potentially leaving some areas uncovered. (3) For regional targets spanning large and irregularly oriented ranges of latitude and longitude, the coverage efficiency of a single RGT orbit remains suboptimal.

In contrast to the aforementioned methods, this chapter incorporates the orbital parameters of candidate RGT orbits as decision variables within the constellation design model. This expands the decision space for satellite placement, enabling superior design outcomes with fewer satellites. First, a satellite-target coverage analysis model is established based on the theory of cyclic convolution, allowing for the efficient modeling of coverage performance for regional targets. Subsequently, the constellation design model is extended to accommodate multiple RGT orbits, leading to the proposal of a Simultaneous Orbit

Optimization and Satellite Deployment (SOOSD) model. This model facilitates the co-optimization of constellation orbital parameters and satellite deployment positions. To tackle the model's high-dimensional decision variables, sparse target coverage, and non-linear constraints, a two-stage optimization framework is designed, integrating a differential evolution algorithm with an integer programming solver. The significant advantages of the proposed method in terms of satellite count, coverage performance, and computational efficiency are validated through several simulation cases with diverse geographical distributions and coverage requirements.

3.2 Satellite-Target Coverage Analysis based on RGT Orbits

3.2.1 Satellite Orbit Dynamics and Orbit Representation

The motion of a satellite around the Earth is primarily governed by gravitational attraction. However, it is also subject to various perturbing forces, including the non-spherical gravity of the Earth, third-body gravity, atmospheric drag, and solar radiation pressure. For LEO satellites, the most significant perturbations are from the Earth's non-spherical gravity and atmospheric drag. Since atmospheric drag can cause a rapid decay in orbital altitude, this dissertation assumes that its effects are compensated for through regular orbit maintenance. Consequently, atmospheric drag is neglected in the orbital model. Among the non-spherical gravitational effects, the second-zonal harmonic, denoted as J_2, has the most dominant influence. Therefore, this chapter considers only the J_2 perturbation. The orbital dynamics of a satellite under the influence of the J_2 perturbation can be described in the Earth-Centered Inertial (ECI) frame as [109]:

$$
\begin{cases}
\ddot{x} = -\frac{\mu x}{r^3}\left[1 + \frac{3}{2}J_2\left(\frac{R_e}{r}\right)^2\left(1 - 5\frac{z^2}{r^2}\right)\right] \\
\ddot{y} = -\frac{\mu y}{r^3}\left[1 + \frac{3}{2}J_2\left(\frac{R_e}{r}\right)^2\left(1 - 5\frac{z^2}{r^2}\right)\right] \\
\ddot{z} = -\frac{\mu z}{r^3}\left[1 + \frac{3}{2}J_2\left(\frac{R_e}{r}\right)^2\left(3 - 5\frac{z^2}{r^2}\right)\right]
\end{cases}
\tag{3.1}
$$

where $\boldsymbol{r} = [x, y, z]^{\mathrm{T}}$ is the position vector of the spacecraft relative to the ECI frame, $r = \|\boldsymbol{r}\|$ is the distance from the satellite to the origin of ECI frame, μ is the Earth's gravitational constant, and R_e is the Earth's radius.

To describe the characteristics of a satellite's orbit more intuitively, the six classical orbital elements are commonly used: semi-major axis (a), eccentricity (e), inclination (i), right ascension of the ascending node (RAAN, Ω), argument of perigee (ω), and true anomaly (f). The semi-major axis a determines the size of the orbit. The eccentricity e defines the shape of the orbit. The inclination i is the angle between the orbital plane and the Earth's equatorial plane. The RAAN Ω is the angle from the vernal equinox to the ascending node. The argument of perigee ω is the angle from the ascending node to the perigee. The true anomaly f specifies the position of the satellite in its orbit. These six orbital elements uniquely define a satellite's orbit. For computational convenience, the mean anomaly M and eccentric anomaly E are often used instead of the true anomaly f to represent the satellite's position. The relationships between them are as follows:

$$
M = E - e \sin E \tag{3.2}
$$

$$
E = 2 \arctan\left(\sqrt{\frac{1-e}{1+e}} \tan \frac{f}{2}\right) \tag{3.3}
$$

For near-circular orbits, where $e \to 0$, these three anomalies are approximately equal.

However, as $e \to 0$, the argument of perigee becomes singular. Since LEO constellations often use near-circular orbits, the argument of latitude $u = \omega + f$ can be used to represent the satellite's position in the orbit. The satellite's position and velocity vectors can be converted to and from the six orbital elements, as detailed in [109].

Under the influence of the J_2 perturbation, the secular rates of change for the RAAN, argument of perigee, and mean anomaly are given by [110]:

$$\begin{cases} \dot{\Omega} = -\frac{3}{2}\frac{J_2 R_e^2}{p^2} n \cos i \\ \dot{\omega} = \frac{3}{4}\frac{J_2 R_e^2}{p^2} n \left(4 - 5\sin^2 i\right) \\ \dot{M} = n\left[1 + \frac{3}{4}\frac{J_2 R_e^2}{p^2}\sqrt{1-e^2}\left(2 - 3\sin^2 i\right)\right] \end{cases} \tag{3.4}$$

where $p = a(1-e^2)$ is the semi-latus rectum, and $n = \sqrt{\mu/a^3}$ is the mean motion.

3.2.2 Repeat Ground Track Orbit

A Repeat Ground Track (RGT) orbit is one in which a satellite passes over the same point on the Earth's surface after a fixed period. RGT orbits are commonly employed in the design of regional coverage constellations to ensure that satellites can cover specific areas within a certain timeframe [28], [47], [48]. The orbital period of an RGT orbit must satisfy the following relationship:

$$N_s T_s = N_e T_e \tag{3.5}$$

where T_s is the satellite's orbital period around the Earth, and T_e is the Earth's rotation period relative to the orbit plane. N_s and N_e are coprime positive integers, representing the number of satellite revolutions and Earth rotations, respectively, within one repeat cycle.

When considering the J_2 perturbation, the design of an RGT orbit must account for the effects of Earth's rotation and the precession of the satellite's orbital plane. In this case, T_s and T_e are given by:

$$T_s = \frac{2\pi}{\dot{\omega} + \dot{M}} \tag{3.6}$$

$$T_e = \frac{2\pi}{\omega_e - \dot{\Omega}} \tag{3.7}$$

where ω_e is the Earth's angular rotation rate. The period ratio τ of the RGT orbit is defined as [28]:

$$\tau = \frac{N_s}{N_e} = \frac{T_e}{T_s} = \frac{\dot{\omega} + \dot{M}}{\omega_e - \dot{\Omega}} \tag{3.8}$$

The value of τ is typically predetermined by the constellation designer based on coverage requirements. For a near-circular orbit, given τ and the orbital inclination i, the semi-major axis a can be determined by solving the following equation, which combines Eq. (3.4) and Eq. (3.8):

$$\sqrt{\frac{\mu}{a^3}}\left[1 - \frac{3J_2R_e}{2a^2}\left(1 - 4\cos^2 i + \frac{N_s}{N_e}\cos i\right)\right] = \frac{N_s}{N_e}\omega_e \tag{3.9}$$

The semi-major axis a in the above equation can be solved using numerical methods.

Figure 3.1 shows a comparison of the ground tracks for a typical RGT orbit and a non-RGT orbit over a two-day period. The green track represents the RGT orbit with parameters $\tau = 4/1$, $e = 0$, $i = 50°$, $\Omega = 0$, $\omega = 0$, and $f = 0$. The semi-major axis is calculated as $a = 16727.8$ km using Eq. (3.9). The red track represents the non-RGT orbit with parameters $\Omega = 50°$ and $a = 16927.8$ km, while other parameters are the same as the RGT orbit. As can be seen, the ground track of the RGT orbit repeats on the second day, consistent with the set value of τ, whereas the non-RGT orbit's track does not.

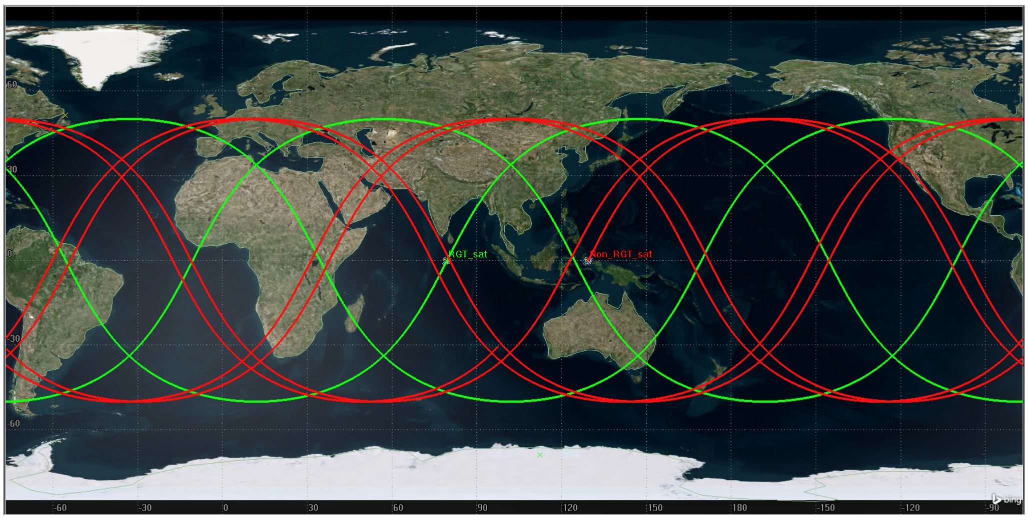


Figure 3.1: Comparison of RGT orbit and non-RGT orbit.

3.2.3 Satellite-to-Target Communication Coverage Condition

The problem of designing continuous coverage constellations for regional targets primarily focuses on LEO communication constellations. In this context, the coverage area of a satellite on the ground at any given moment is determined by the field of view of the satellite's communication payload and the minimum elevation angle of the ground user. This chapter assumes that the satellite's communication payload has sufficient power and capacity to meet the access demands of all users within the coverage area, thus only geometric coverage conditions need to be considered. This assumption is consistent with common practices in related constellation design literature [28], [34], [35], [50].

According to Homssi [111], the satellite's coverage on the ground is determined by the smaller of the satellite's and the ground target's beamwidths. This chapter assumes that the satellite's beamwidth is large enough to cover up to the tangent of the Earth. Therefore, the effective beamwidth is mainly determined by the ground target's beamwidth. The beamwidth of a ground target is typically determined by its minimum elevation angle ε.

The ground target's beamwidth ψ_t can be expressed as: $\psi_t = \pi - 2\varepsilon$. As shown in Figure 3.2, let ψ be the angle subtended by the satellite's effective coverage area on the ground. ψ can be calculated by the following equation:

$$\psi = 2 \arcsin\left(\frac{H}{\|\boldsymbol{r}_s(t)\|} \cos \varepsilon\right), \varepsilon > 0. \tag{3.10}$$

where $H = R_e + h$ is the distance from the ground target to the Earth's center, and h is the altitude of the ground target. $\|\boldsymbol{r}_s(t)\|$ is the satellite's orbital radius at time t. The maximum geocentric angle corresponding to the satellite's coverage area is [111]:

$$\theta_m = 2 \arcsin\left(\frac{\|\boldsymbol{r}_s(t)\|}{H} \sin \frac{\psi}{2}\right) - \psi. \tag{3.11}$$

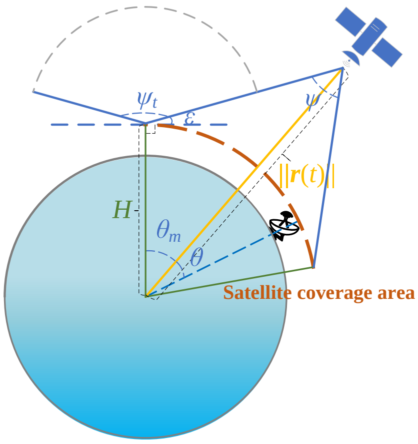


Figure 3.2: Diagram of coverage conditions of satellites for ground targets.

Defining the contact angle between the satellite and the target as the geocentric angle $\theta(\boldsymbol{r}_s(t), \boldsymbol{r}_t(t))$, the condition for a satellite to cover a target can be described as:

$$\theta(\boldsymbol{r}_s(t), \boldsymbol{r}_t(t)) \leq \frac{\theta_m}{2}. \tag{3.12}$$

where $\boldsymbol{r}_t(t)$ represents the position vector of the target at time t, which can be obtained by:

$$\boldsymbol{r}_t(t) = R(T_0)\begin{bmatrix} \cos\omega_e t & -\sin\omega_e t & 0 \\ \sin\omega_e t & \cos\omega_e t & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} H\cos\phi\cos\lambda \\ H\cos\phi\sin\lambda \\ H\sin\phi \end{bmatrix}, \tag{3.13}$$

where T_0 is the start time of the coverage analysis, and $R(T_0)$ is the transformation matrix from the Earth-Centered, Earth-Fixed (ECEF) frame to the ECI frame at time T_0. Details of the transformation can be found in [110]. ϕ and λ represent the latitude and longitude of the target, respectively.

3.2.4 Coverage Representation

Coverage analysis requires calculating the coverage time intervals between each satellite in the constellation and each target over a given period T. For an RGT orbit, it is sufficient to obtain the coverage time intervals for a single repeat cycle ($T_{\text{RGT}} = N_s T_s$) to derive long-term coverage analysis results. To facilitate the representation of the coverage analysis results, the entire period can be discretized into L time slots with a fixed time interval t_{step}. Then, within each time slot, the coverage status of the satellite for the target is determined based on the coverage condition described in Section 3.2.3. To describe the coverage of satellite k for point target j, a satellite coverage indicator vector $\boldsymbol{v}_{kj} = \{v_{kj}[0], ..., v_{kj}[L-1]\}$ can be defined. The n-th element of $\boldsymbol{v}_{kj}$ indicates whether the target is covered by the satellite during the n-th time slot, and its expression is:

$$v_{kj}[n] = \begin{cases} 1, & \text{if satellite } k \text{ and target } j \text{ satisfy Eq. (3.12) during time slot } n \\ 0, & \text{otherwise} \end{cases} \tag{3.14}$$

where $n \in \{0, 1, ..., L-1\}$ is the index of the time slot.

Correspondingly, the coverage profile for the j-th target can be defined as $\boldsymbol{b}_j = \{b_j[0], ..., b_j[L-1]\}$, with the following expression:

$$b_j[n] = \sum_{k=1}^{K} v_{kj}[n] \tag{3.15}$$

where K is the number of satellites in the same RGT orbit. The value of $b_j[n]$ represents the number of satellites that can access the j-th target during the n-th time slot. In other words, it also reflects the coverage multiplicity of the constellation on that target.

3.2.5 Coverage Calculation Based on Circular Convolution

As shown in Figure 3.3, satellites in an RGT orbit constellation with the same semi-major axis and inclination have identical ground tracks, but different phases along these tracks. Therefore, by randomly selecting a position on the RGT orbit as a reference satellite, the positions of the other satellites can be represented by their phase difference relative to the reference satellite. Compared to describing orbits with six Keplerian elements, this method is more concise, intuitive, and convenient for modeling the constellation design problem.

Assuming an RGT orbit constellation consists of satellites in the same RGT orbit, a constellation pattern vector $\boldsymbol{x} = \{x[0], ..., x[L-1]\}$ can be defined to represent the positions of all satellites on the RGT orbit relative to a reference satellite:

$$x[n] = \begin{cases} 1, & \text{if } n = n_k \\ 0, & \text{otherwise} \end{cases} \tag{3.16}$$

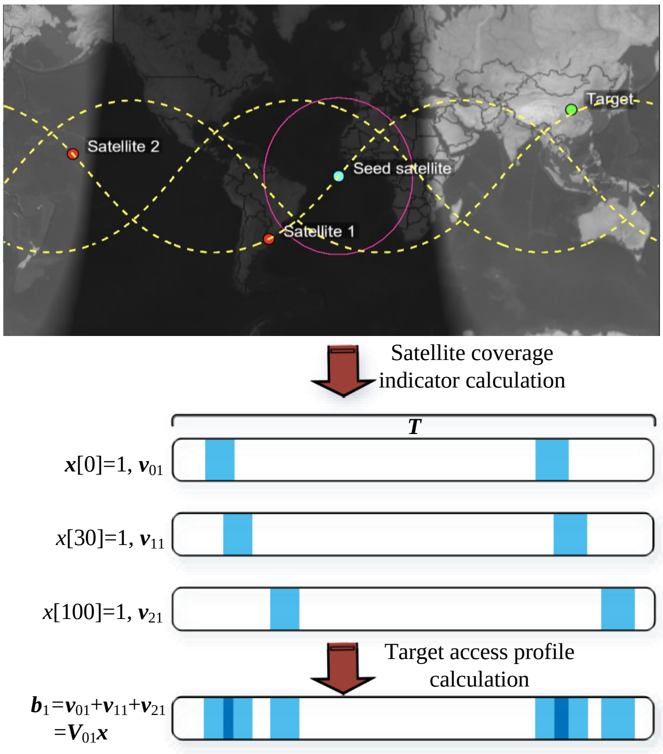


Figure 3.3: Satellite coverage features in the same RGT orbit and target coverage profile calculation.

where n_k is an integer representing the time delay index of satellite k [28]. The above equation means that when $x[n] = 1$, the k-th satellite will "lag" behind the reference satellite on the RGT orbit by a phase difference of $n_k t_{\text{step}}$. This vector can be used to represent the positions of individual satellites in the RGT orbit. Therefore, the number of satellites in the same RGT orbit constellation can be obtained by counting the number of non-zero elements in the constellation pattern vector $\boldsymbol{x}$.

Satellites in the same RGT orbit share the same semi-major axis, orbital inclination, eccentricity, and argument of perigee. The remaining orbital elements (right ascension of the ascending node and mean anomaly) for each satellite can be calculated from the constellation pattern vector and the orbital elements of the reference satellite using the following equations [112]:

$$\Omega_k = n_k \frac{2\pi N_e}{L} + \Omega_0 \tag{3.17}$$

$$[N_s(\Omega_k - \Omega_0) + N_e(M_k - M_0)] \bmod (2\pi) = 0 \tag{3.18}$$

The operator mod$(\cdot)$ represents the modulo operation.

As illustrated in Figure 3.3, the coverage profile $\boldsymbol{b}_j$ for a single target by multiple satellites is typically obtained by summing the coverage indicator vector $\boldsymbol{v}_{kj}$ of each satellite. This approach requires separate orbit propagation for each satellite to determine its coverage of the target, resulting in a computational complexity of $O(L \times K)$. However, as shown in the figure, when satellites are in the same RGT orbit, their coverage profiles $\boldsymbol{b}_j$ for the same target have the same form, differing only in phase. Therefore, they can be treated as the same circular signal with different phases. By leveraging the circular convolution property from digital signal processing, the target's coverage profile $\boldsymbol{b}_j$ can be efficiently calculated by performing a circular convolution of the constellation pattern vector $\boldsymbol{x}$ and the reference satellite's coverage indicator vector $\boldsymbol{v}_{0j}$. Specifically, $\boldsymbol{b}_j$ can be expressed as [28]:

$$\boldsymbol{V}_{0j}\boldsymbol{x} = \boldsymbol{b}_j \tag{3.19}$$

where $\boldsymbol{V}_{0j}$ can be calculated by:

$$V_{0j}[\alpha, \beta] = v_{0j}[(\alpha - \beta) \bmod L] \tag{3.20}$$

α and β represent the row and column indices within $\boldsymbol{V}_{0j}$, respectively. Using this circular convolution property, the complexity of single-target coverage analysis can be reduced to $O(L)$.

Algorithm 3-1 Coverage analysis

Require: RGT orbit seed satellite parameter: I, Ω, τ; Minimum elevation angle of airplane ϵ; Coverage analysis start time T_0; RGT repetition period T; RGT constellation pattern $\boldsymbol{x}$; time interval t_{step}; target set J.

Ensure: Coverage profile for each target $\boldsymbol{b}_j$.

Set $e = 0$, $M = 0$;
Calculate a by Eq. (3.9);
$L = \lceil T/t_{\text{step}} \rceil$;
Calculate the satellite position $\boldsymbol{r}_s(T_0)$ at time T_0 based on the seed satellite orbit parameters;
for $j = 1$ to $|J|$ **do**
    for $n = 1$ to L **do**
        $t = T_0 + (n - \frac{1}{2})t_{\text{step}}$;
        Calculate satellite position $\boldsymbol{r}_s(t)$ by Eq. (3.1);
        Calculate j-th target position $\boldsymbol{r}_{tj}(t)$ by Eq. (3.13);
        Calculate $\theta(\boldsymbol{r}_s(t), \boldsymbol{r}_{tj}(t))$;
        if Eq. (3.12) is satisfied **then**
            $v_{0j}[n] = 1$;
        else
            $v_{0j}[n] = 0$;
        end if
    end for
    Calculate $\boldsymbol{V}_{0j}$ by Eq. (3.20);
    Calculate $\boldsymbol{b}_j$ by Eq. (3.19);
end for

The coverage of a regional target can be approximated by discretizing it into a set of point targets. The coverage for the entire region is then represented by the collective coverage profile of these individual points. This set of point targets can be obtained by uniformly partitioning the regional target. Furthermore, the coverage profile for the set of point targets can be extended from Eq. (3.19) as follows:

$$\begin{bmatrix} \boldsymbol{V}_{01} \\ \boldsymbol{V}_{02} \\ \cdots \\ \boldsymbol{V}_{0|J|} \end{bmatrix} \boldsymbol{x} = \begin{bmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \\ \cdots \\ \boldsymbol{b}_{|J|} \end{bmatrix} \tag{3.21}$$

where $J = \{[\phi_1, \lambda_1, h_{p1}], ..., [\phi_{|J|}, \lambda_{|J|}, h_{|J|}]\}$ is the set of point targets, and $|J|$ denotes the number of elements in J, i.e., the number of point targets.

Algorithm 3-1 presents the pseudocode for conducting a coverage analysis for a set of point targets using a constellation in the same RGT orbit, based on the circular convolution property described above. By performing a single orbit propagation and one circular convolution operation, the coverage profile for a single point target can be obtained. Consequently, the computational complexity can be reduced from $O(L \times |J| \times K)$ to $O(L \times |J|)$.

3.3 Regional Coverage Constellation Design Model Based on RGT Orbits

This section first presents a satellite deployment model for a single RGT orbit. It then extends this model to a multi-RGT orbit scenario, proposing the Simultaneous Orbit Optimization and Satellite Deployment (SOOSD) model. As this chapter primarily focuses on the coverage performance of satellite constellations for complex regional targets, the modeling is based on the following assumptions:

Assumption 3.1. *Only the J_2 perturbation is considered among non-spherical gravita-*

tional effects on satellite motion.

Assumption 3.2. *A sufficient number of ground stations are available to provide continuous satellite access, ensuring that a communication link can be established between a user and a ground station whenever the ground target is covered by a satellite.*

Assumption 3.3. *The communication capacity of each satellite is large enough to consistently meet the demands of all users within its coverage area.*

Assumption 3.4. *To ensure uniform communication quality across different ground regions, the satellite orbits are assumed to be near-circular.*

3.3.1 Satellite Deployment Model Based on a Single RGT Orbit

We first consider the regional coverage constellation design problem for a single RGT orbit. This problem involves determining the specific position of each satellite within a given RGT orbit to minimize the number of satellites while satisfying the spatio-temporal coverage requirements for a regional target, which is represented by a set of point targets. In this model, the RGT orbit's period ratio τ, inclination i, RAAN Ω, and mean anomaly M are all pre-defined. The regional coverage constellation design problem can be formulated as a satellite deployment model based on a single RGT orbit, as described in [28]:

(1) Decision Variables

1) $\boldsymbol{x}$: The constellation pattern vector for the RGT orbit.

The constellation pattern vector $\boldsymbol{x}$ represents the positions of all satellites in the RGT orbit relative to a reference satellite. The length of this vector is L, where L is the number of time slots in one repeat cycle. Once the value of $\boldsymbol{x}$ is determined, the RAAN and mean anomaly for each satellite can be obtained using Eq. (3.17) and Eq. (3.18), respectively.

(2) Objective

$$\min \|\boldsymbol{x}\|_1 \tag{3.22}$$

The objective function (3.22) minimizes the total number of satellites, and thus the total constellation cost, by minimizing the L1 norm of the constellation pattern vector. In principle, the number of satellites is the count of non-zero elements in $\boldsymbol{x}$ (the L0 norm). However, based on the definition of $\boldsymbol{x}$ in Eq. (3.16), minimizing the L1 norm also achieves the goal of minimizing the total satellite count.

(3) Constraints

$$\boldsymbol{V}_{0j}\boldsymbol{x} \geq \boldsymbol{f}_j, \ \forall j = 1, ..., |J| \tag{3.23}$$

$$x[n] \in \{0, 1\}, \ \forall n = 0, ..., L-1 \tag{3.24}$$

Constraint (3.23) ensures that the coverage profile for each target meets the specified coverage requirements. Here, $\boldsymbol{f}_j$ is a user-defined requirement vector, where each element specifies the minimum number of accessible satellites required for the target during the corresponding time slot. Constraint (3.24) enforces the binary nature of the decision variables.

This satellite deployment model is a linear integer programming problem, which can be solved directly using commercial solvers such as CPLEX [1] or Gurobi [2]. However, due to the high dimensionality of the decision variables and the sparsity of the matrix $\boldsymbol{V}_{0j}$, obtaining a globally optimal solution within a reasonable time is often challenging. As a trade-off, a suboptimal feasible solution can be obtained within a limited solving time (e.g., one hour).

[1] https://www.ibm.com/products/ilog-cplex-optimization-studio
[2] https://www.gurobi.com/solutions/gurobi-optimizer/

3.3.2 Simultaneous Orbit Optimization and Satellite Deployment Model Based on Multi-RGT Orbits

The satellite deployment model based on a single RGT orbit has the following limitations: (1) The RGT orbit parameters are pre-defined, which may lead to low coverage efficiency for each satellite, thus requiring more satellites to meet the coverage demand. In practice, orbital parameters should be optimized based on the location of the target area. (2) For scenarios with small satellite coverage footprints (determined by the minimum elevation angle ε and the satellite's semi-major axis a) and widely distributed regional targets, a single RGT orbit may fail to provide effective coverage for all targets, potentially leaving some areas uncovered. (3) For regional targets spanning large and irregularly oriented ranges of latitude and longitude, the coverage efficiency of a single RGT orbit remains suboptimal. Since the ground tracks of satellites in the same RGT orbit maintain a consistent direction over a specific region, multiple RGT orbits are necessary to accommodate regional targets with diverse orientations.

To address these shortcomings, this section proposes the Simultaneous Orbit Optimization and Satellite Deployment (SOOSD) model. This model allows satellites to be deployed in multiple RGT orbits and simultaneously optimizes the parameters of each orbit. This approach enables efficient coverage of complex regional targets and further reduces the required number of satellites while satisfying coverage demands. The mathematical formulation of the SOOSD model is as follows [113]:

(1) Decision Variables

In the SOOSD model, all RGT orbits share the same period ratio τ, which is set by the user based on coverage requirements. According to the near-circular orbit assumption, the eccentricity and argument of perigee for all RGT orbits are set to zero. The mean anomaly of the reference satellite for each RGT orbit is also set to zero. Thus, the decision variables for the SOOSD model are:

1) i_k: The orbital inclination of the k-th RGT orbit.
2) Ω_k: The RAAN of the k-th RGT orbit.
3) $\boldsymbol{x}_k$: The constellation pattern vector for the k-th RGT orbit.

In the variables above, $k = 1, ..., N$ is the index of the RGT orbit, where N is the number of available RGT orbits. The constellation pattern vector $\boldsymbol{x}_k$ for each orbit is similar to $\boldsymbol{x}$ in the single-RGT-orbit model, representing the positions of satellites in the k-th RGT orbit relative to its reference satellite. Unlike the single-orbit model, the SOOSD model allows for multiple RGT orbits, thus requiring a separate constellation pattern vector for each.

(2) Objective

$$\min \sum_{k=1}^{N} \|\boldsymbol{x}_k\|_1 \tag{3.25}$$

Similar to Eq. (3.22), the objective function (3.25) aims to minimize the total number of satellites across all RGT orbits.

(3) Constraints

$$i_{\text{min}} \leq i_k \leq i_{\text{max}}, \ \forall k = 1, \dots, N \tag{3.26}$$

$$0 \leq \Omega_k \leq \frac{360°}{\tau}, \ \forall k = 1, \dots, N \tag{3.27}$$

$$\boldsymbol{V}_{0j}^{k} = f_{\text{ca}}(\tau, i_k, \Omega_k, \varepsilon, T_0, T_{\text{RGT}}), \ \forall k = 1, \dots, N, \ j = 1, \dots, |J| \tag{3.28}$$

$$\sum_{k=1}^{N} \boldsymbol{V}_{0j}^{k} \boldsymbol{x}_k \geq \boldsymbol{f}_j, \ \forall j = 1, \dots, |J| \tag{3.29}$$

$$x_k[n] \in \{0, 1\}, \ \forall k = 1, \dots, N, \ n = 0, \dots, L-1 \tag{3.30}$$

Constraint (3.26) defines the range for the orbital inclination. $i_{\min}$ is set to the minimum inclination required to cover the target with the highest latitude. Since orbits with inclinations i_k and $(180° - i_k)$ have similar ground tracks, this chapter sets $i_{\max}$ to $90°$. Constraint (3.27) specifies the range for the RAAN of each RGT orbit. According to the properties of RGT orbits, orbits with RAAN values of $\{\Omega_k + m\frac{360°}{\tau},\ m = 1, 2, ...\}$ have identical ground tracks. To improve solution efficiency, the range of RAAN is restricted to $[0, \frac{360°}{\tau}]$. Constraint (3.28) indicates that the coverage matrix of the reference satellite for each target depends on decision variables such as inclination and RAAN. The function $f_{ca}(\cdot)$ represents the target coverage analysis process described in Algorithm 3-1. The presence of (3.28) makes the SOOSD model a Mixed-Integer Nonlinear Programming (MINLP) problem. Constraint (3.29) ensures that the sum of coverage profiles provided by each sub-constellation (satellites within the same RGT orbit) meets the target's coverage requirements. Constraint (3.30) enforces the binary nature of the decision variables.

3.4 Two-Stage Solving Framework Combining Differential Evolution and Integer Programming Solver

This section introduces the two-stage solution framework designed to solve the SOOSD model. It begins by outlining the design philosophy of the framework, followed by a description of the differential evolution-based bi-level optimization method used in the first stage to efficiently determine the optimal orbital parameters for each candidate RGT orbit.

3.4.1 Design of the Solving Framework

Due to the presence of constraint (3.29), the SOOSD model is classified as a complex MINLP problem. A common approach for solving such intricate MINLP problems is to use metaheuristic intelligent optimization algorithms, such as PSO [114], GA [115], and Differential Evolution (DE) [116]. These algorithms offer high flexibility, do not require gradient information, and can readily handle complex constraints and optimization objectives that are difficult to model. However, the sparsity and high dimensionality of constraint (3.29) make it challenging to achieve satisfactory results when directly optimizing high-dimensional binary decision variables with these algorithms. Conversely, commercial solvers like Gurobi can rapidly find feasible solutions for binary integer programming problems, such as the one described in Section 3.3, with theoretical guarantees on solution optimality. Therefore, by leveraging the respective strengths of these two approaches, the performance of solving the SOOSD model can be significantly enhanced.

Furthermore, an analysis of the SOOSD model reveals that if constraint (3.28) is disregarded, the model can be decomposed into two sub-problems: an RGT orbit parameter optimization model (defined by (3.26) and (3.27)) and a satellite deployment model based on multiple RGT orbits (defined by (3.25), (3.29), and (3.30)). These two sub-problems are coupled through the coverage analysis process, as indicated by (3.28). Based on the above analysis, this chapter proposes a divide-and-conquer-based two-stage solution framework, as illustrated in Figure 3.4.

In the first stage, a DE algorithm is employed as the primary algorithm to iteratively optimize the orbital parameters i_k and Ω_k for each RGT orbit. Within each iteration, it is necessary to solve the multi-RGT-orbit satellite deployment sub-problem to determine the optimal number of satellites for the current set of orbital parameters. However, solving this

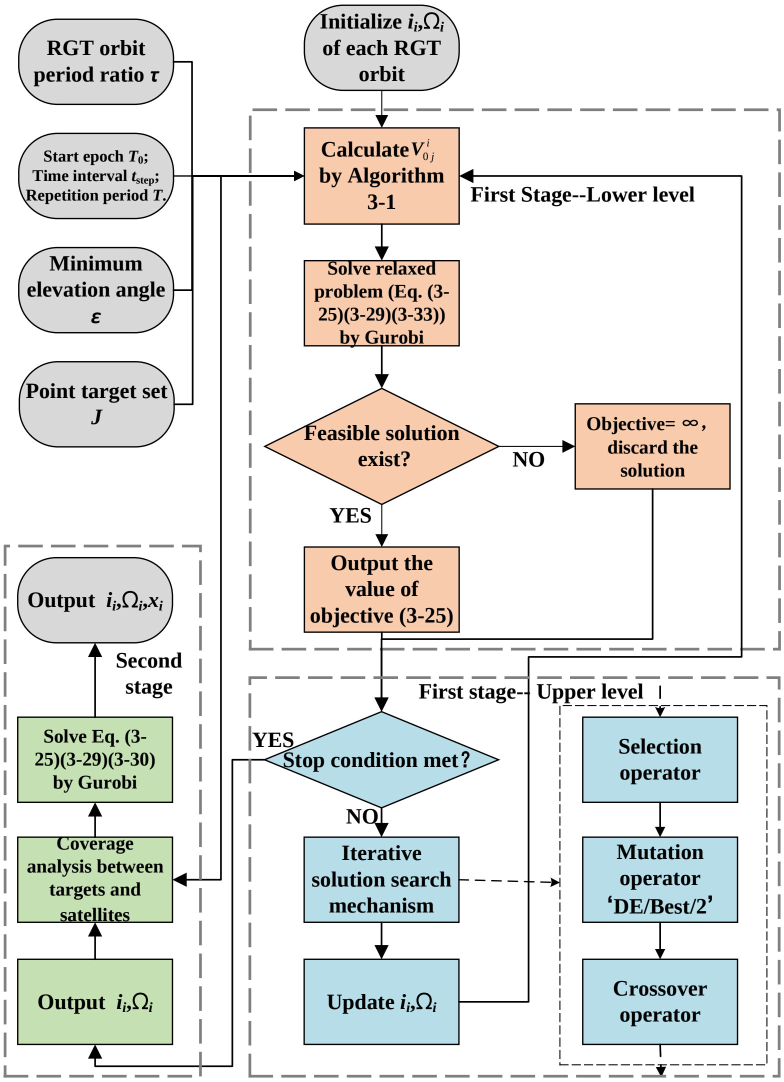


Figure 3.4: Two-stage solving framework for the constellation design problem based on RGT orbit [113].

sub-problem is typically time-consuming. Repeatedly solving the integer programming model across thousands of DE iterations would lead to an unacceptably long total solution time. To address this, this chapter proposes solving a linear relaxation of the original multi-RGT-orbit satellite deployment model in the lower level. Compared to solving the integer programming model directly, the solution time for the linear relaxation is substantially reduced. Although the objective value from the linear relaxation does not represent the actual number of satellites required, it serves as a lower bound to the objective of the original integer problem. This lower bound effectively reflects the trend of the original objective value, thereby guiding the iterative optimization of the orbital parameters.

In the second stage, the optimal orbital parameters for each RGT orbit, obtained from the first stage, are used as inputs to determine the coverage vector for each target. Subsequently, a multi-RGT-orbit satellite deployment sub-model with binary variables is formulated and solved using Gurobi. This process yields the final optimal satellite deployment scheme.

3.4.2 Differential Evolution-Based Bi-Level Optimization Method

Differential Evolution (DE) [117] is a population-based optimization algorithm adept at solving complex optimization problems with continuous decision variables. The algorithm maintains a set of potential solutions, known as the population, where each individual is represented as a vector in the search space. Assuming the parent population is $P_{\text{parent}} = \left[\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_{N_p}\right]$, where N_p is the population size, $\boldsymbol{p}_i$ is the i-th individual, and $\boldsymbol{p}_i = \left[p_i^1, p_i^2, ..., p_i^D\right]$ represents a candidate solution for all decision variables, with D being the length of the individual. The initial population is randomly generated using the following formula [118]:

$$
\begin{aligned}
p_i^j &= lb^j + \text{rand}(0,1) \cdot (ub^j - lb^j), \\
& i = 1, 2, \ldots, N_p, \ j = 1, 2, \ldots, D
\end{aligned}
\tag{3.31}
$$

where lb^j and ub^j are the lower and upper bounds of the j-th decision variable, respectively.

In the upper-level optimization of this problem, the DE algorithm is used to optimize the orbital inclination i_k and the RAAN Ω_k for each RGT orbit. Therefore, $D = 2N$, where the first N positions encode the inclination of each RGT orbit, and the subsequent N positions encode the RAAN. The bounds lb^j and ub^j for each position are determined by (3.26) and (3.27).

As shown in Figure 3.4, after the initial population is generated, the upper-level algorithm iteratively updates the population through three main operations: mutation, crossover, and selection. Apart from fitness evaluation, the main iterative process of DE includes the following four parts:

(1) Mutation. In the mutation process, DE generates a mutant vector by randomly selecting and combining several different individuals. This chapter employs the "DE/best/2" mutation operator, the specific form of which can be found in [117].

(2) Crossover. The crossover operation combines the mutant individual with a parent individual to create a trial individual. Its expression is:

$$
u_i^j = \begin{cases} \nu_i^j, & \text{if } \text{rand}_i^j(0,1) \le p_{\text{CR}} \\ p_i^j, & \text{otherwise} \end{cases}
\tag{3.32}
$$

where p_{CR} is the crossover probability, which is preset by the user and remains constant throughout the iterations. u_i^j is the j-th decision variable of the i-th trial individual, and ν_i^j is the j-th decision variable of the i-th new individual obtained from the mutation operation.

(3) Selection. In the selection operation, the trial individual is compared with its parent individual. If the fitness of the trial individual is better than that of the parent, the trial individual replaces the parent in the next generation. The fitness function serves as the basis for this selection.

(4) Fitness Evaluation. As depicted in Figure 3.4, fitness evaluation corresponds to the lower-level process and consists of two main parts. First, based on the candidate orbital inclination and RAAN for each RGT orbit, a coverage analysis is performed to obtain the target coverage vector for each reference satellite. Second, the linearly relaxed multi-RGT-orbit satellite deployment model is solved. This model is composed of (3.25) and (3.28), along with the following relaxed decision variable constraint:

$$0 \leq x_k[n] \leq 1, \ \forall k = 1, \ldots, N, \ n = 0, ..., L-1 \tag{3.33}$$

When the linear relaxation model can find a global optimum within a specified time limit, the lower level returns the objective value as the fitness value for the upper-level DE algorithm to guide the iteration. If the solver fails to find a feasible solution within the time limit or if the model is infeasible, the fitness value is set to ∞.

3.5 Simulation of Case Study

This section presents a simulation case study of constellation design based on RGT orbits for three regional airway targets with different shapes and geographical locations to validate the effectiveness and superiority of the proposed model and solution framework. First, Section 3.5.1 provides detailed settings for the three regional targets. Subsequently, Sections 3.5.2-3.5.4 present the constellation design results for the three cases. In Case 1, a comparative analysis is conducted with other constellation design methods, such as the Street-of-Coverage, and the computational efficiency of the proposed two-stage solution framework is compared with that of direct metaheuristic algorithms. The relevant comparison results are presented in Section 3.5.2[3]. Finally, to further verify the superiority of the SOOSD model and the designed solution framework, a comparison is made with a constellation design method that only considers satellite deployment optimization, based on the scenarios from the literature [28]. All simulations were conducted on a laptop equipped with an Intel Core i7-12700H CPU and 16 GB of RAM.

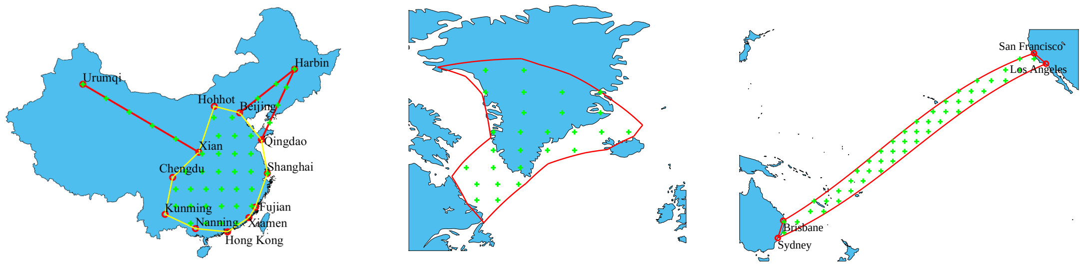
3.5.1 Scenario and Simulation Parameter Settings

The regional airway targets are generated based on flight data from the website Flightradar24[4]. Three regional airway targets with different shapes, scales, and geographical locations were designed, as shown in Figure 3.5. The detailed data for the three types of regional airway targets are as follows:

- Case 1: Major airways over mainland China, covering the central and southeastern coastal areas, composed of 11 cities and three boundary airways. This target is uniformly divided into 42 point targets to test the coverage performance of the algorithm for complex-shaped targets at mid-latitudes.
- Case 2: Airway corridor over Greenland, formed by 21 boundary points. This regional target is divided into 26 point targets to test the algorithm's coverage perfor-

[3] Airway target and detailed constellation design results can be found in: https://github.com/HITPengHan/RGT-consteallation-design

[4] www.flightradar24.com



(a) Case 1 (major airways over Chinese mainland)

(b) Case 2 (airways over Greenland Island)

(c) Case 3 (airways connecting Oceania and North America)

Figure 3.5: Geographical distribution of the regional airway targets and their split point target representations.

mance for high-latitude regional targets.

- Case 3: Airway corridor between Australia and North America, connecting Sydney, Brisbane, San Francisco, and Los Angeles. This target is divided into 46 point targets to test the algorithm's coverage performance for long, narrow regional targets spanning large ranges of latitude and longitude.

Other parameters related to the scenario, reference satellite orbit, ground targets, and algorithm are set as shown in Table 3.1.

3.5.2 Simulation Results and Analysis of Single-Fold Coverage for a Mid-Latitude Complex-Shaped Target

3.5.2.1 Constellation Design and Flight Test Results

The constellation design results for Case 1 were obtained using the two-stage solution framework proposed in this chapter, based on the parameter settings described above. The specific results are shown in Table 3.2 and Figures 3.6 to 3.10.

Figure 3.6 shows the optimization iteration process of the first stage based on DE. The algorithm converges to the optimal value after approximately 500 Number of Fitness Eval-

Table 3.1: Simulation parameter settings.

Parameter	Case 1	Case 2	Case 3
RGT orbit period ratio τ		12/1	
Inclination range $[i_{\min}, i_{\max}]$	$[11°, 90°]$	$[39.5°, 90°]$	$[0°, 90°]$
RAAN range		$[0, 30°]$	
Mean anomaly of reference satellite M		$0°$	
Start time T_0		1/1/2000 12:00:00	
Minimum elevation angle of airplane ε	$40°$	$30°$	$30°$
Airplane altitude h		10 km	
Repeat period time step t_{step}		120 s	
Maximum number of RGT orbits N		3	
Coverage requirement	$\boldsymbol{f}_j = \mathbf{1}$,	$\boldsymbol{f}_j = \mathbf{1}$,	$\boldsymbol{f}_j = \mathbf{2}$, $j = 1, ..., \lvert J \rvert$.
Crossover and mutation probabilities for DE		[0.8,0.1]	
Population size for DE		30	
Number of fitness evaluations for DE		1200	
Maximum runtime for Gurobi		1 hour	

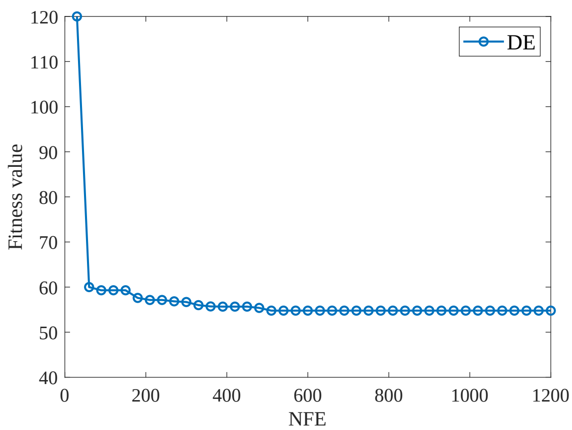
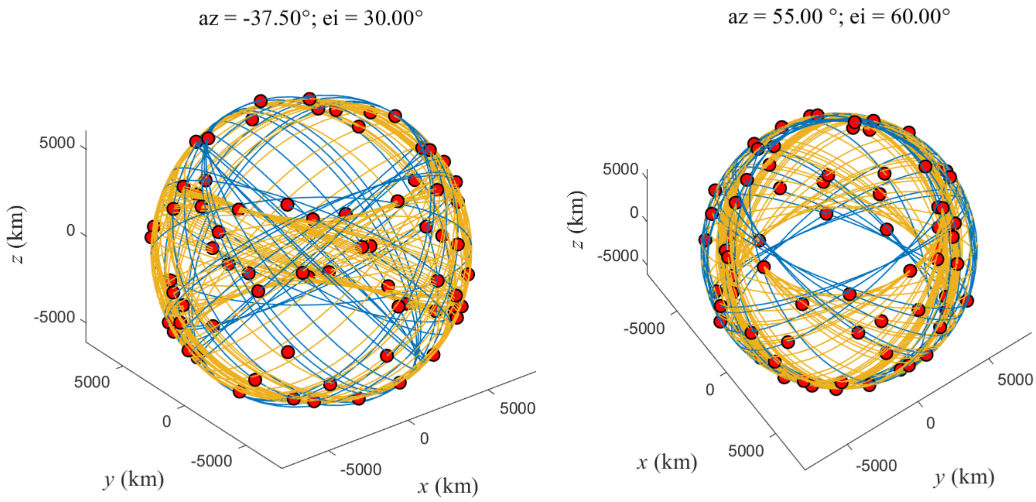


Figure 3.6: Iteration process for a single run of DE.

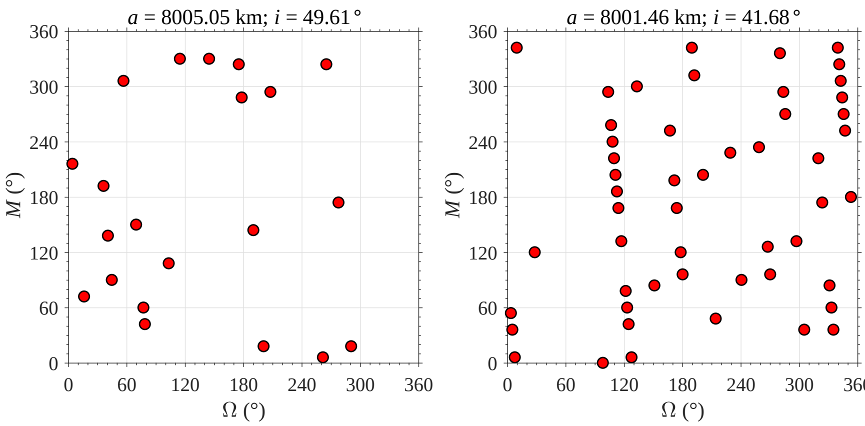
Table 3.2: Constellation design results of Case 1

Constellation Type	RGT Orbit No.	RGT Orbit seed Satellite Parameters Semi-major Axis/km	Inclination/°	RAAN/°	Satellites in RGT Orbit	Total Satellites
Multi-RGT Orbit	1	8005.05	49.61	21.50	21	**72**
	2	8001.52	41.68	7.45	51	
Single-RGT Orbit	1	8002.62	44.47	13.18	111	111

The other orbital parameters (eccentricity, argument of perigee, and mean anomaly) are all set to 0.



(a) Satellite initial position and orbit distribution in 3D space



(b) Satellite distribution in $\Omega - M$ frame

Figure 3.7: Satellite distribution of constellation design results of Case 1.

uations (NFE), which indicates that the set number of NFE is reasonable and validates the convergence performance of the algorithm. Figure 3.7 illustrates the satellite distribution in the designed constellation, including the 3D spatial distribution and the distribution in the $\Omega - M$ frame. These satellites are distributed on two of the three candidate RGT orbits. The satellite trajectories in the first RGT orbit are shown in blue, and those in the second are in yellow. From Figure 3.7(a), it can be observed that each satellite occupies a unique orbital plane to ensure consistent ground tracks. In Figure 3.7(b), the satellites in the two RGT orbits are represented in two separate subplots. It can be seen that each satellite has a different RAAN, which also results in each satellite occupying an independent orbital plane.

Table 3.2 provides the detailed parameters and number of satellites for each RGT orbit in both the multi-RGT orbit and single-RGT orbit schemes. It can be seen that the constellation design based on a single RGT orbit requires 111 satellites. In contrast, the multi-RGT orbit scheme results in 72 satellites distributed across two RGT orbits, reducing the required number of satellites by nearly 35% compared to the single-RGT orbit scheme. Furthermore, combining with Fig. 3.7, it can be observed that the inclination and RAAN of the two RGT orbits are different, leading to different ground track directions over the target area. The RGT orbit with a lower inclination accommodates more satellites to cover the extensive mid-to-low latitude areas, whereas the RGT orbit with a higher inclination uses fewer satellites to cover the high-latitude boundary regions. For regional targets with complex shapes and wide coverage, this multi-RGT orbit design significantly enhances satellite coverage efficiency, thereby reducing the number of satellites required and verifying the effectiveness and superiority of the proposed SOOSD model.

To further validate the coverage performance of the designed constellation, eight flights originating from different cities were introduced into the design scenario, and their access

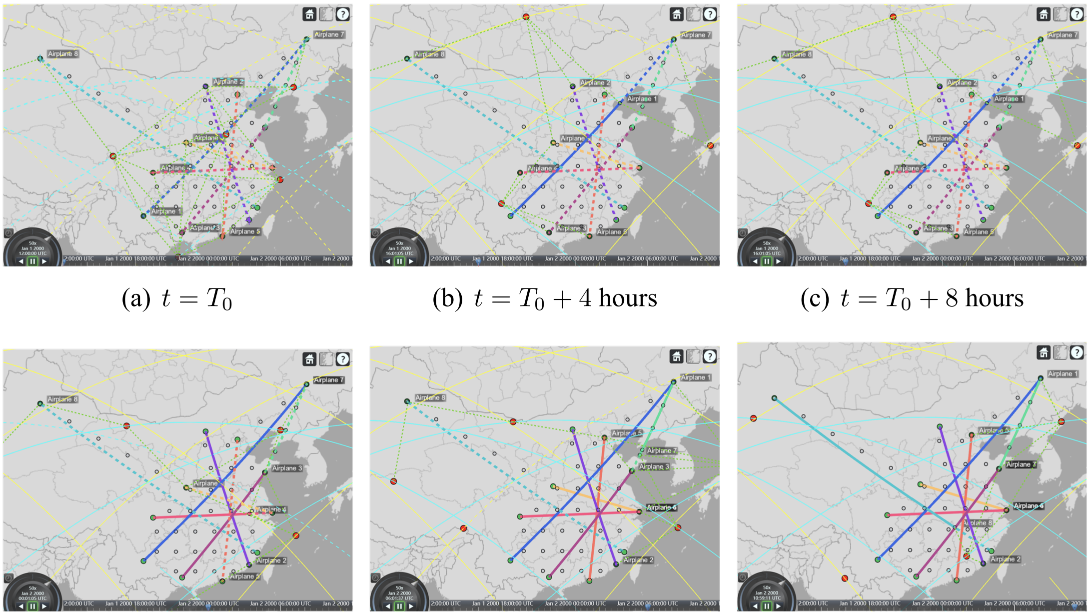
Table 3.3: Settings of the eight flights in Case 1.

Flight No.	Departure City	Arrival City	Start Time	Flight Duration
1	Kunming	Harbin	1/1/2000 12:00	6 hours
2	Hohhot	Xiamen	1/1/2000 16:00	4 hours
3	Nanning	Qingdao	1/1/2000 20:00	2.5 hours
4	Chengdu	Shanghai	1/1/2000 22:00	2.5 hours
5	Hong Kong	Beijing	1/2/2000 0:00	3.5 hours
6	Xi'an	Shanghai	1/2/2000 3:00	2 hours
7	Harbin	Qingdao	1/2/2000 4:00	2.5 hours
8	Urumqi	Fuzhou	1/2/2000 6:00	6 hours

status with the constellation was evaluated. The detailed information for each flight is provided in Table 3.3. These flights connect all boundary points of the regional airway target in Case 1 and cover the main part of this target area. The flight duration of each flight is consistent with the actual flight duration from Filghtradar24[5]. The departure time of each flight was specifically set to ensure that it covers the entire constellation repeat period, thereby comprehensively measuring the constellation's coverage performance. It should be noted that actual flight paths are complex and have a degree of randomness. To simplify the analysis, this chapter approximates the actual flight path using the shortest straight-line distance between the departure and arrival cities.

Figure 3.8 shows the satellite-to-airplane connection status at different times and locations. The six subplots indicate that all eight airplanes can access at least one satellite at different times and locations, demonstrating that the satellites in the constellation can achieve full coverage of the regional airway target. Figure 3.9 further presents the accessible time intervals for each airplane with each satellite during its flight. It can be seen that each accessible time interval is relatively short, thus requiring dozens of satellites to achieve continuous coverage of the target. Figure 3.10 shows the statistical results of the

[5] www.flightradar24.com



(a) $t = T_0$ (b) $t = T_0 + 4$ hours (c) $t = T_0 + 8$ hours

(d) $t = T_0 + 12$ hours (e) $t = T_0 + 18$ hours (f) $t = T_0 + 23$ hours

Figure 3.8: Satellite connectivity status of eight flights at different instants in Case 1.

number of accessible satellites for each airplane during its flight. All eight subplots indicate that each airplane can always access at least one satellite during its flight, meeting the expected coverage requirements. These results all validate the reasonableness of the designed constellation.

3.5.2.2 Comparison with Other Constellation Design Methods

This section provides a more comprehensive comparative analysis of the proposed multi-RGT orbit deployment scheme, the single-RGT orbit scheme, and the Street-of-Coverage scheme [39] under different values of ε. Furthermore, for the Street-of-Coverage scheme, three coverage modes were tested: latitude band coverage, polar cap coverage, and global coverage. The latitude range for the latitude band coverage was set to $23° - 40°$, which is perfectly aligned with the latitude range of the selected regional airway target. The semi-

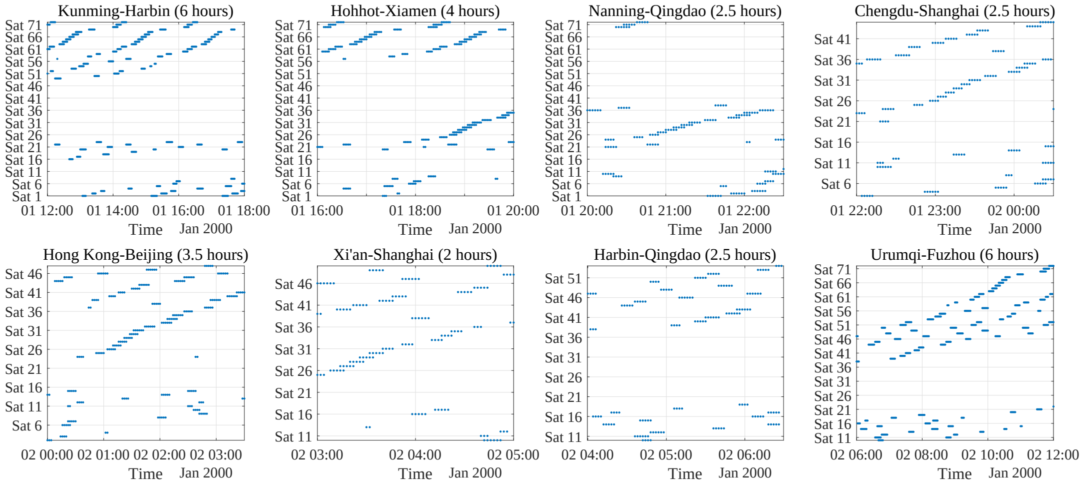


Figure 3.9: Satellite-to-airplane accessible timeline during the flying of the eight flights in Case 1.

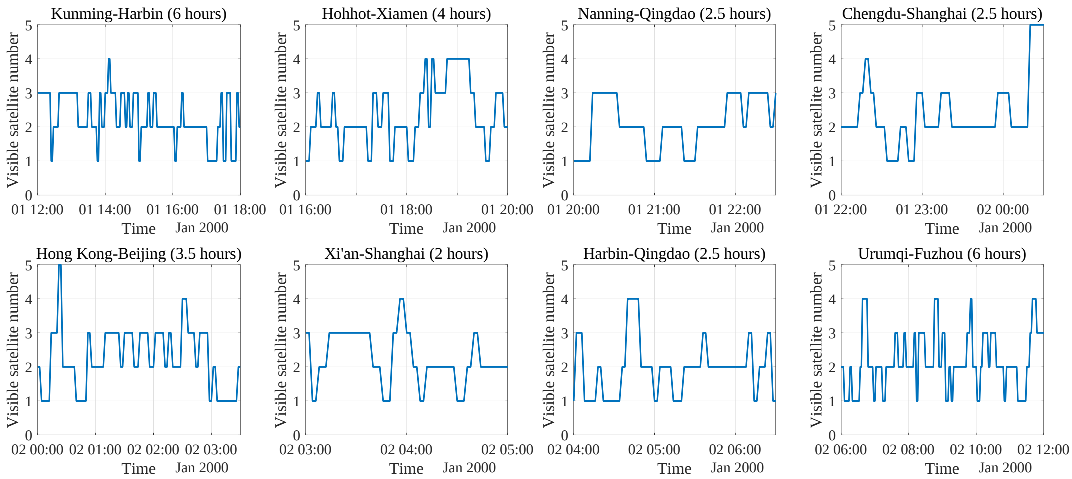


Figure 3.10: Accessible satellite number count during the flying of the eight flights in Case 1.

major axis of the Street-of-Coverage constellation was set to 8005 km, which is almost identical to that of an RGT orbit with a period ratio of 12/1. The minimum elevation angle ε was varied from $10°$ to $60°$ in steps of $10°$. The remaining parameters were kept consistent with those in Section 3.5.1. The comparison results for the two types of schemes are shown in Table 3.4.

As can be seen, the RGT orbit-based method requires a comparable or smaller number of satellites than the Street-of-Coverage method in most cases. The only exception is at $\varepsilon = 20°$, where the Street-of-Coverage method with latitude band coverage performs better than the RGT orbit method. Furthermore, as ε increases, the gap in the number of required satellites between the two types of constellation design methods widens. Specifically, when $\varepsilon = 60°$, the multi-RGT orbit scheme can reduce the number of satellites by up to 47.6% compared to the Street-of-Coverage latitude band coverage scheme. For polar cap and global coverage scenarios, the number of required satellites increases significantly due to the much larger coverage area compared to regional and latitude band coverage.

Table 3.4: Comparison results of different constellation design methods under different ε in Case 1.

Minimun Elevation Angle ε	Number of Satellites				
	Street-of-Coverage [39]			RGT Orbit	
	Latitude Band	Polar Cap	Global	Single-RGT	Multi-RGT
$10°$	24	52	80	24	**24**
$20°$	**36**	88	140	41	41
$30°$	68	160	238	67	**54**
$40°$	114	280	420	111	**72**
$50°$	240	504	780	192	**191**
$60°$	533	1008	2070	360	**279**

Moreover, by comparing the results of the multi-RGT orbit scheme with the single-RGT orbit scheme, it is found that when the minimum elevation angle ε is $10°$ and $20°$, both schemes require the same number of satellites. This is because a lower ε value significantly expands the accessible range of a single satellite, allowing it to completely cover the regional airway target. In this case, a single RGT orbit can achieve the optimal solution. However, as ε increases, the multi-RGT orbit scheme consistently achieves a smaller number of satellites. Across different values of ε, the multi-RGT orbit scheme reduces the number of satellites by an average of about 15% compared to the single-RGT orbit scheme. This is because when ε increases, leading to a smaller satellite coverage area that cannot cover the entire airway target, the single-RGT orbit-based constellation design scheme becomes inefficient. These results further demonstrate that the multi-RGT orbit constellation design method, combined with orbit parameter optimization, has higher design efficiency for regional targets spanning large ranges of latitude and longitude.

Additionally, it can be observed that as ε increases linearly, the number of satellites required by all constellation design schemes grows exponentially. Therefore, to reduce the deployment and launch costs of the constellation, it is crucial to enhance the communication payload capabilities of satellites to achieve broader communication coverage.

3.5.2.3 Comparison with Other Algorithms

This section compares the proposed two-stage solution framework with methods that directly solve the SOOSD model. The comparison is intended to investigate whether the two stage method is necessary in solving the proposed SOOSD model. The comparison methods include several classic population-based metaheuristic algorithms, such as Genetic Algorithm (GA) [115], Differential Evolution (DE) [117], and Success-History based Adaptive Differential Evolution (SHADE) [119]. The termination condition for all algorithms was set to a maximum runtime of 4 hours. The population size for each algorithm was tuned by testing population sizes of 300, 500, and 1000 on the constellation design problem with $\varepsilon = 30°$. The results showed that the optimal population sizes for GA, DE, and SHADE were 500, 500, and 300, respectively. The crossover and mutation probabilities for GA and DE were also tuned, with the final optimal values being [0.9, 0.5] for GA and [0.6, 0.5] for DE. All three algorithms were implemented based on the evolutionary optimization platform PlatEMO [120]. With the optimal parameter settings and the same termination condition, the three direct-solving heuristic algorithms were tested under different values of ε.

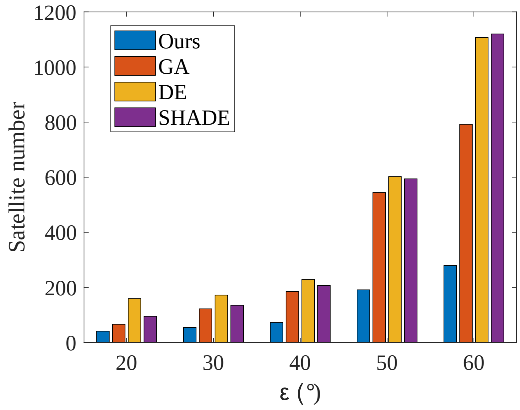


Figure 3.11: Comparison results between the two-stage solution framework and direct-solving algorithms under different ε settings in Case 1.

The comparison simulation results are shown in Figure 3.11. It is evident that the two-stage solution framework proposed in this chapter achieves a significantly smaller number of satellites in all scenarios. Compared to the other algorithms, the proposed two-stage optimization method achieves a reduction of 38%-75% in the number of satellites. Furthermore, comparing the optimization results of each algorithm at $\varepsilon = 60°$ and $\varepsilon = 20°$, our method shows an increase in satellite count of about 5.8 times, while the direct-solving methods show an increase of about 6-11 times. This demonstrates the robustness of our method to changes in ε.

This is because metaheuristic optimization algorithms tend to get stuck in local optima or even fail to find a feasible solution when directly solving optimization problems with a large number of binary variables. In this chapter, we address this by linearly relaxing the satellite deployment problem with binary variables and embedding it into the fitness evaluation process of the metaheuristic optimization algorithm. The decision variables in the DE algorithm are only the parameters of each RGT orbit, thus avoiding the direct optimization of the binary variables x using a metaheuristic algorithm. After obtaining the optimal RGT orbit parameters from the linearly relaxed satellite deployment model, the second stage of the algorithm framework directly uses the commercial solver Gurobi, which is adept at solving integer programming problems, to solve the original satellite deployment model and obtain the final constellation design. Therefore, the algorithm demonstrates good robustness and superiority when solving problems of different scales.

3.5.3 Simulation Results and Analysis of Single-Fold Coverage for a High-Latitude Target

Table 3.5 presents the constellation design results for Case 2, including the parameters and number of satellites for each RGT orbit. For the multi-RGT orbit scheme, a total of three RGT orbits were used. The first RGT orbit has a near-90° inclination with only 3 satellites deployed, while the majority of satellites are distributed on the other two RGT orbits with inclinations of about 70°. The total number of satellites for the multi-RGT orbit scheme is 48, which is still better than the 49 satellites required by the single-RGT orbit scheme. The satellite distribution in the constellation is shown in Figure 3.12. In Figure 3.12(a), the satellite distribution in 3D space is clearly visible, with the satellite trajectories of the three RGT orbits represented by blue, yellow, and green, respectively.

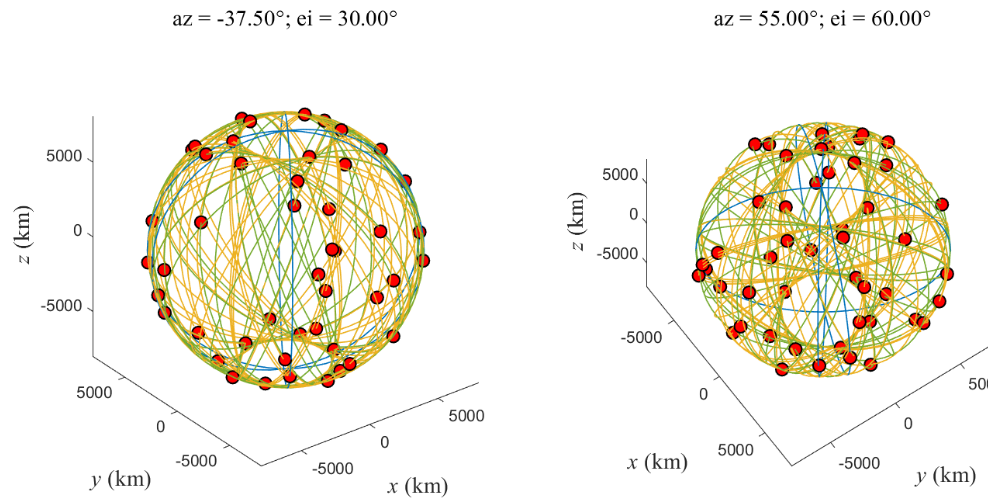
Figure 3.12(b) shows the satellite distribution in the $\Omega - M$ frame, where the number of satellites on each RGT orbit is consistent with Table 3.5.

Table 3.5: Constellation design results of Case 2.

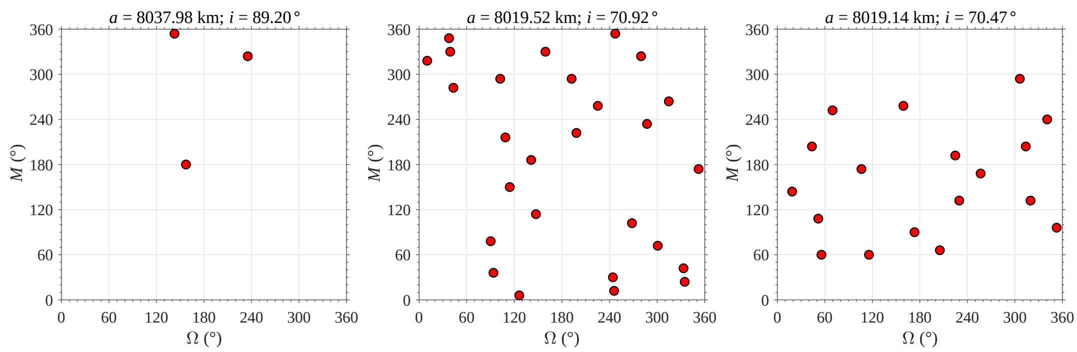
Constellation Type	RGT Orbit No.	RGT Orbit Seed Satellite Parameters			Satellites in RGT Orbit	Total Satellites
		Semi-major Axis/km	Inclination/°	RAAN/°		
Multi-RGT Orbit	1	8037.97	89.20	21.26	3	**48**
	2	8019.52	70.92	6.76	27	
	3	8019.14	70.47	0.75	18	
Single-RGT Orbit	1	8020.67	72.25	7.05	49	49

The other orbital parameters (eccentricity, argument of perigee, and mean anomaly) are all set to 0.

To further validate the coverage performance of the designed constellation, four different flights within the target area were set up, and their satellite access status during flight was evaluated. These flights all depart from and arrive at the boundary points of the regional airway target, with different start and end times to ensure coverage of the entire constellation repeat period. The detailed information for each flight is provided in Table 3.6. The access evaluation results are shown in Figures 3.13 and 3.14. The results indicate that due to the short duration of the flights, the airplanes cannot access all satellites during their flight. However, accessing only a portion of the satellites is sufficient to ensure that all four flights are continuously covered by at least one satellite throughout their entire flight, meeting the single-fold continuous coverage requirement. This result further validates the adaptability of the proposed method for high-latitude regional airway targets.



(a) Satellite initial position and orbit distribution in 3D space



(b) Satellite distribution in $\Omega - M$ frame

Figure 3.12: Satellite distribution of constellation design results of Case 2.

3.5.4 Simulation Results and Analysis of 2-Fold Coverage for a Narrow Target

The constellation design results for Case 3 are shown in Table 3.7 and Figure 3.15. It can be seen that the multi-RGT orbit scheme uses a total of two RGT orbits, with a total of 110 satellites. In comparison, the single-RGT orbit scheme requires 120 satellites, about 9% more than the multi-RGT orbit scheme. The vast majority of satellites in the multi-RGT

Table 3.6: Settings of the four flights in Case 2.

Flight No.	Departure City	Arrival City	Start Time	Flight Duration
1	P2 [75.30°N, 60.19°W]	P15 [68.48°N, 13.99°W]	1/1/2000 12:00	5 hours
2	P15 [68.48°N, 13.99°W]	P7 [60.13°N, 63.78°W]	1/1/2000 18:00	5.5 hours
3	P8 [55.08°N, 59.66°W]	P15 [68.48°N, 13.99°W]	1/2/2000 00:00	5 hours
4	P3 [72.95°N, 55.90°W]	P13 [64.66°N, 23.22°W]	1/2/2000 5:00	4 hours

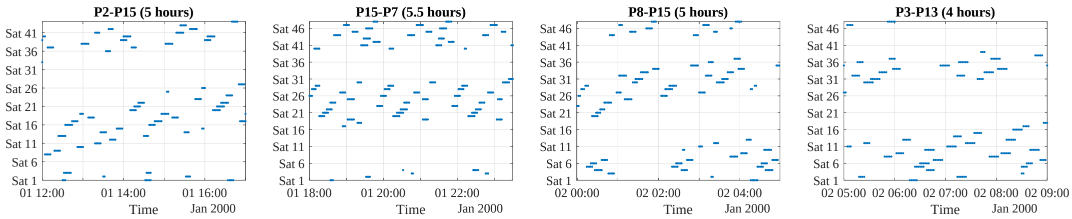


Figure 3.13: Satellite-to-airplane accessible timeline during the flying of the four flights in Case 2.

orbit scheme are distributed in RGT orbit 1, and the orbital parameters of RGT orbit 1 are similar to the single-RGT orbit result. However, by adding only two extra satellites to RGT orbit 2, the 2-fold coverage requirement is met while significantly reducing the total number of satellites. Due to the large span of latitude and longitude of this regional target and the 2-fold coverage requirement in Case 3, the number of required satellites is significantly higher than in the previous two cases. From Figure 3.15(b), it can be seen that the satellites of the designed constellation are approximately uniformly distributed on RGT orbit 1.

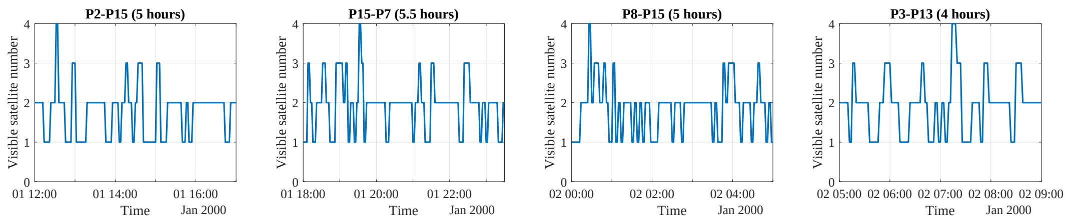
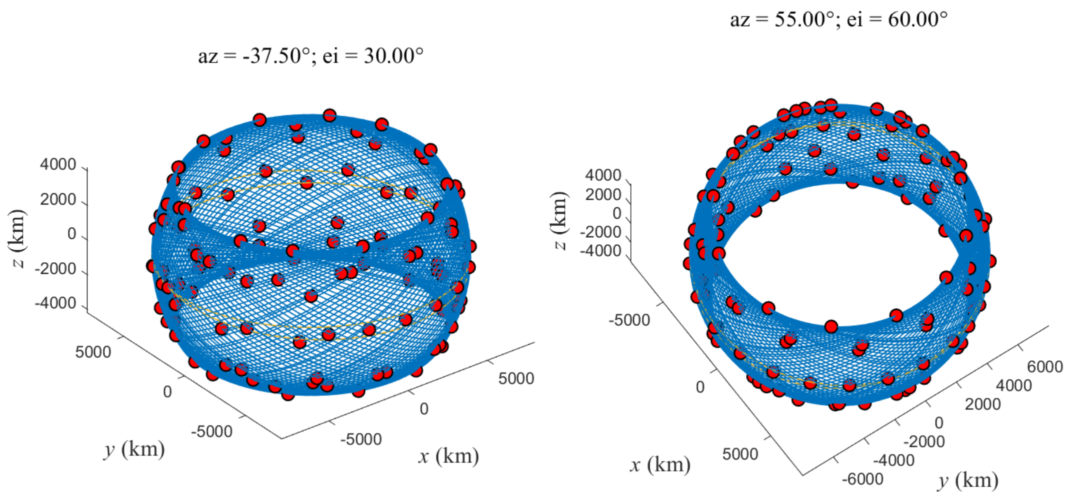


Figure 3.14: Accessible satellite number count during the flying of the four flights in Case 2.

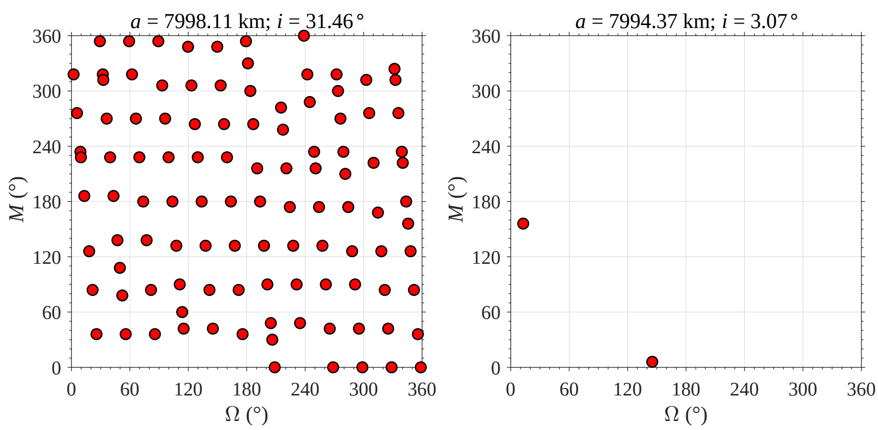
Table 3.7: Constellation design results of Case 3.

Constellation Type	RGT Orbit No.	RGT Orbit Seed Satellite Parameters			Satellites in RGT Orbit	Total Satellites
		Semi-major Axis/km	Inclination/°	RAAN/°		
Multi-RGT Orbit	1	7998.10	31.46	28.73	108	**110**
	2	7994.36	3.07	25.82	2	
Single-RGT Orbit	1	7997.76	29.79	1.434	120	120

The other orbital parameters (eccentricity, argument of perigee, and mean anomaly) are all set to 0.



(a) Satellite initial position and orbit distribution in 3D space



(b) Satellite distribution in $\Omega - M$ frame

Figure 3.15: Satellite distribution of constellation design results of Case 3.

To validate the coverage performance of the designed constellation, two flights originating from different regions were set up, with details provided in Table 3.8. The simulation results are shown in Figures 3.16 and 3.17. Both airplanes can access at least two satellites throughout their entire flight, fully meeting the 2-fold continuous coverage design requirement. Furthermore, due to the long flight duration and large geographical span of the two flights, almost all satellites in the constellation can establish a connection with the airplanes during their flight. Similar to the previous cases, no loss of connection with the constellation occurred during the flight. This case further validates the applicability of the constellation design method proposed in this chapter for regional targets with large latitude and longitude spans and multi-fold coverage requirements.

Table 3.8: Settings of the two flights in Case 3.

Flight No.	Departure City	Arrival City	Start Time	Flight Duration
1	Sydney	San Francisco	1/1/2000 12:00	14 hours
2	Los Angeles	Brisbane	1/1/2000 21:00	15 hours

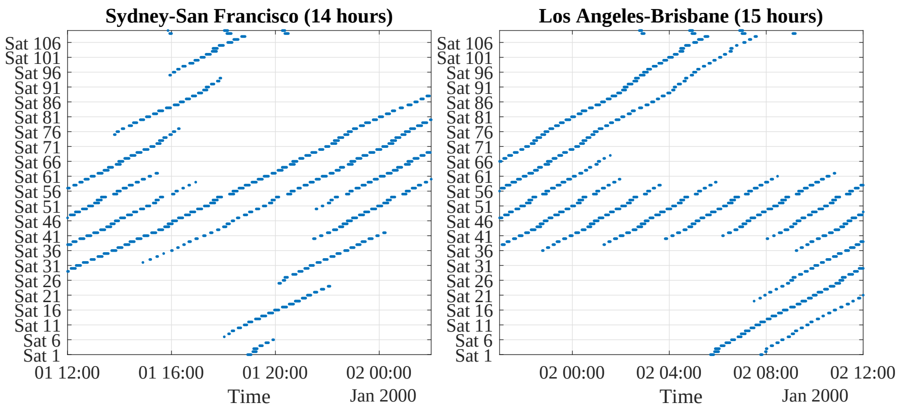


Figure 3.16: Satellite-to-airplane accessible timeline during the flying of the two flights in Case 3.

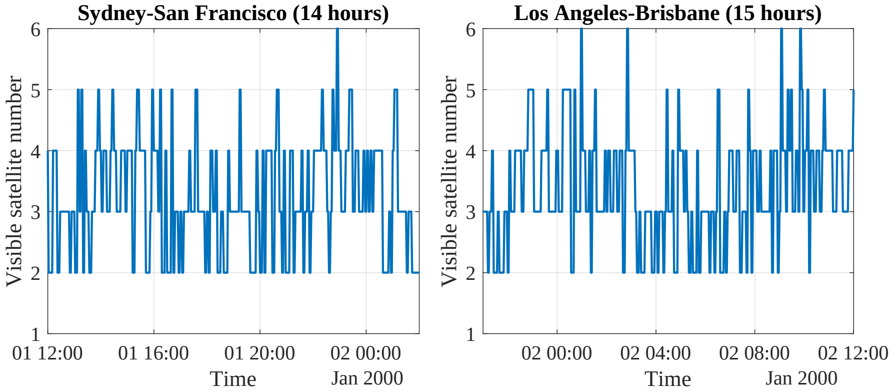


Figure 3.17: Accessible satellite number count during the flying of the two flights in Case 3.

3.5.5 Comparison with Results in Existing Literature

The work in [28] also studied regional target coverage design based on RGT orbits. To further validate the universality and superiority of the model and solution framework proposed in this chapter, three typical cases from [28] were selected for a comparative study. All three cases require single-fold, full-time continuous coverage for the selected targets. The specific parameter settings are shown in Table 3.9. In each case, the proposed two-stage solution framework was used to jointly optimize the orbital elements of the RGT orbit reference satellite and the satellite deployment positions to minimize the required number of satellites. The order of orbital elements in the table is: repeat period, eccentricity, inclination, argument of perigee, RAAN, and mean anomaly.

Table 3.9: Case settings in [28].

No.	RGT Orbit Reference Satellite Elements	ε	Ground Target Set	Coverage Req.	L
1	$[12/1, 0, 102.9°, 0°, 98.3°, 0°]^{\mathrm{T}}$	5°	{[34.75°N, 84.35°W]}	$\boldsymbol{f}_j = 1$	720
3	$[5/1, 0.41, 63.435°, 90°, 0°, 0°]^{\mathrm{T}}$	30°	{Antarctica}		718
5	$[8/1, 0, 70°, 0°, 0°, 0°]^{\mathrm{T}}$ $[6/1, 0, 47.915°, 0°, 0°, 0°]^{\mathrm{T}}$	$\varepsilon_1 = 15°$ $\varepsilon_2 = 10°$	{[64.14°N, 21.94°W], [19.07°N, 72.87°E]}		717

In the first case, while keeping the repeat period of the RGT orbit reference satellite unchanged, the inclination and RAAN were treated as decision variables. Since the target is a single point, only one candidate RGT orbit was set. The optimization results are shown in Table 3.10. It can be seen that by jointly optimizing the inclination and RAAN of the reference satellite, a better constellation configuration can be obtained than in [28] (the required number of satellites was reduced from 18 to 14).

Table 3.10: Result comparison of Case 1 in [28].

Source	Reference Satellite Elements	Optimal Deployment	Sat. Count
[28]	$[12/1, 0, 102.9°, 0°, 98.3°, 0°]^{\mathrm{T}}$	$x^*[n] = \begin{cases} 1, \text{for } n = 39; 73; 79; 89; 170; 184; 234; 250; 331; 341; 347; 492; 502; 542; 638; 648; 654; 663. \\ 0, \text{otherwise} \end{cases}$	18
This Chapter	$[12/1, 0, 49.08°, 0°, 67.36°, 0°]^{\mathrm{T}}$	$x^*[n] = \begin{cases} 1, \text{for } n = 332; 342; 352; 362; 372; 382; 392; 659; 669; 679; 689; 699; 709; 719. \\ 0, \text{otherwise} \end{cases}$	**14**

In the third case, the eccentricity, inclination, argument of perigee, and RAAN of the RGT orbit reference satellite were treated as decision variables, as this RGT orbit is elliptical. At the same time, three candidate RGT orbits were set to expand the search space. The optimization results are shown in Table 3.11. By adjusting the orbital elements of the reference satellite, our method obtained a constellation configuration with only 3 satellites, which is better than the 5 satellites result in [28]. Furthermore, all 3 satellites are located on the same RGT orbit. Compared to the reference orbit in the original paper, the optimized reference RGT orbit has a larger eccentricity and inclination.

In the fifth case, while keeping the parameter settings from [28] unchanged, the same number of two candidate RGT orbits as in the original paper were set. At the same time, the repeat period of each reference satellite was kept constant, and the inclination and RAAN were treated as decision variables, consistent with the first case. The results in

Table 3.11: Result comparison of Case 3 in [28].

Source	Reference Satellite Elements	Optimal Deployment	Sat. Count
[28]	$[5/1, 0.41, 63.435°, 90°, 0°, 0°]^{\mathrm{T}}$	$x^*[n] = \begin{cases} 1, \text{for } n = 96; 310; 358; \\ 562; 612. \\ 0, \text{ otherwise} \end{cases}$	5
This Chapter	$[5/1, 0.5112, 92.03°, 0°, 6.81°, 0°]^{\mathrm{T}}$	$x^*[n] = \begin{cases} 1, \text{for } n = 633; 693; 719. \\ 0, \text{ otherwise} \end{cases}$	**3**

Table 3.12 show that our method again obtained a better result with only 8 satellites (the original paper required 10). Furthermore, unlike the configuration in [28] which required two RGT orbits with different repeat periods, our results show that these satellites only need to be deployed on the same RGT orbit to meet the coverage requirements.

Table 3.12: Result comparison of Case 5 in [28].

Source	Reference Satellite Elements	Optimal Deployment	Sat. Count
[28]	$[8/1, 0, 70°, 0°, 0°, 0°]^{\mathrm{T}}$	$x^*_{(1)}[n] = \begin{cases} 1, \text{for } n = 65; 144; 285; \\ 361. \\ 0, \text{otherwise} \end{cases}$	10
	$[6/1, 0, 47.195°, 0°, 0°, 0°]^{\mathrm{T}}$	$x^*_{(2)}[n] = \begin{cases} 1, \text{for } n = 208; 428; 523; \\ 608; 634; 702. \\ 0, \text{ otherwise} \end{cases}$	
This Chapter	$[6/1, 0, 64.85°, 0°, 6.92°, 0°]^{\mathrm{T}}$	$x^*[n] = \begin{cases} 1, \text{for } n = 12; 37; 195; \\ 225; 371; 399; 548; 581. \\ 0, \text{ otherwise} \end{cases}$	**8**

The three cases above cover continuous coverage constellation design problems for single-point, multi-point, and high-latitude regional targets. In all these cases, the method proposed in this chapter outperformed the results in [28], with all results achieved under optimized orbital parameter configurations different from the original paper. This fully demonstrates the necessity of optimizing RGT orbit parameters during the constellation design process and validates the superiority and wide applicability of the designed two-stage solution method.

Table 3.13: Runtime comparison.

Case No.	Two-Stage Framework (This Chapter)			Method in [28]
	Stage 1	Stage 2	Total Time	
1	7314.5 s	36.7 s	7351.2 s	5937.6 s
3	18545.2 s	87.3 s	18632.5 s	748.4 s
5	2743.4 s	80.2 s	2823.6 s	5298.7 s

Table 3.13 provides a comparison of the runtime between our two-stage solution framework and the method in [28]. It should be noted that in all cases, Gurobi terminated early after reaching the global optimum, without reaching the set maximum runtime. It can be seen that the total time of our two-stage framework is higher than the method in [28] for Case 1 and Case 3, but lower in Case 5. Essentially, the second stage of our two-stage framework and the method in [28] solve the same type of satellite deployment problem, and both use the Gurobi solver. However, in all three test cases, the runtime of the second stage of our framework is significantly shorter than that of [28]. The reason is that the metaheuristic algorithm in the first stage can screen out candidate RGT orbits with better coverage for the target. Particularly in test Case 5, the total runtime of the two-stage framework is even less than that of the method in [28], while achieving a better result. Overall, the two-stage solution framework is inferior in computational efficiency to the method in [28], but it consistently obtains better results by optimizing the RGT orbit positions. In the preliminary design phase of a constellation, a better result is often more important than a shorter computation time.

3.6 Summary

This chapter presents a design method of continuous coverage constellations for regional targets using RGT orbits. First, by leveraging the unique properties of RGT orbits, a target coverage analysis method based on circular convolution is introduced. Second, a multi-RGT orbit Simultaneous Orbit Optimization and Satellite Deployment (SOOSD) model is developed. This model facilitates the co-optimization of RGT orbit parameters and satellite deployment positions, aiming to minimize the required number of satellites while ensuring continuous coverage. To address the challenges posed by the non-linearity of the target coverage coefficients and coverage analysis constraints, which make the problem difficult to solve, a two-stage optimization framework is proposed. This framework integrates a DE algorithm with a branch-and-bound-based integer programming solver. Finally, the effectiveness and superiority of the proposed solution framework are validated through several simulation cases with diverse coverage requirements. Comparative simulation results against single-RGT-orbit schemes and the global-coverage Street-of-Coverage design method demonstrate that the multi-RGT-orbit-based constellation design can significantly reduce the required number of satellites. Compared to direct optimization methods such as GA and SHADE, the proposed framework reduces the satellite count by 38%-75%. The proposed method was also tested on the case study from the literature [28], and the results obtained were superior to those reported in [28], thereby validating the necessity of orbit parameter optimization within the SOOSD model.

Chapter 4

Bi-Objective Optimization of OTV-Based Launch and Deployment Process for LEO Constellation

4.1 Introduction

The cost of constellation construction includes not only the development and manufacturing but also significant expenses in the launch process, such as the research, development, manufacturing, transportation, and ground support for launch vehicles. Launch costs typically account for a substantial portion of the constellation construction budget, especially for large-scale LEO constellation deployment missions, where the number of launches directly impacts the overall economic viability. Therefore, there is an urgent need to optimize the launch and deployment (L&D) process by rationally planning the satellite combination and deployment sequence for each launch. This optimization aims to reduce the required number of launches and improve the utilization of launch vehicle resources, thereby effectively controlling the overall cost of the constellation, which is of great significance for constellation construction.

Current research on constellation L&D primarily focuses on direct and indirect injection methods, with optimization variables typically limited to the payload combination for a single launch, injection orbit parameters, launch site selection, and satellite allocation. Some literature also explores deployment strategies utilizing the J_2 perturbation effect for orbital transfers, but their operational mechanisms and optimization requirements are fundamentally different from OTV-based L&D. The OTV-based L&D strategy, due to its reliance on high-precision, multi-target orbital maneuvers, leads to a significant increase in fuel consumption, which introduces new challenges to the L&D optimization problem. A feasible approach is to model the deployment process as a MSSRP problem [66]. For MSSRP problems, mainstream methods often employ meta-heuristic intelligent optimization algorithms to optimize the rendezvous sequence and time, aiming to minimize fuel consumption for orbital maneuvers. These algorithms perform well in scenarios with small orbital transfer velocity increments (e.g., in GEO) or where extended transfer times are permissible to leverage the J_2 perturbation effect. However, the OTV-based L&D process considered in this chapter requires deploying satellites to multiple orbital positions (even across orbital planes) in a short period, which significantly increases OTV fuel consumption, and the fuel carried by the OTV is limited. Furthermore, constrained by the launch vehicle's capacity and fairing size, the number of satellites that can be carried in a single launch is also limited. Consequently, these various resource constraints substantially shrink the feasible solution space, causing traditional meta-heuristic algorithms to easily get trapped in local optima or even fail to find feasible solutions [71], [72].

To overcome the aforementioned limitations, this chapter proposes a Cluster and Nearest Sorting initialized Genetic Algorithm with Local Search (CNS-LSGA). This algorithm incorporates two key improvements: (1) a clustering and nearest-neighbor sorting initialization mechanism designed for the problem's characteristics, which can generate high-quality initial solutions within or near the feasible region, effectively avoiding getting stuck with infeasible solutions; (2) the introduction of two local search operators into the main genetic algorithm process to enhance the ability to escape local optima. Additionally, the ϵ-constraint method [121] is combined to handle the bi-objective optimization problem, aiming to fully explore the problem's Pareto front and provide decision-makers with a richer set of L&D schemes. This method transforms the original problem into multiple single-objective subproblems by decomposing the range of one of the objectives. CNS-LSGA solves each subproblem separately, ultimately obtaining the Pareto front solution set.

4.2 Mission Scenario Description

This section describes the proposed L&D process, analyzes the key factors affecting its performance, and subsequently defines the main optimization problem.

(1) LEO Constellation

Let the set of satellites in the constellation to be deployed be $C = \{1, ..., N\}$, where N is the total number of satellites. All satellites are homogeneous, each with a mass of m_s. Each satellite has a unique operational orbital position. To maintain a constant resolution or signal strength over ground targets, LEO satellites typically require a near-circular orbit, which implies $e = 0$ and $\omega = 0$. The operational orbit of the i-th satellite can be represented as $s_i = \{a_i, i_i, \Omega_i, M_i\}$, and the set of all orbital positions is $S = \{s_1, ..., s_N\}$.

(2) OTV-based Satellite L&D Process

This chapter focuses on a rapid L&D strategy based on an Orbital Transfer Vehicle (OTV), which is required to deliver multiple satellites to their respective target orbital positions. Fig. 4.1 illustrates the proposed L&D process. First, a launch vehicle carries several satellites (up to a maximum of $n_{\max}$) along with an OTV to an injection orbit. The inclination and RAAN of the injection orbit are matched with those of the first satellite to be deployed, while its semi-major axis (denoted as a_0) is typically lower than that of the operational orbits. After the separation of the first stage and fairing, the OTV, carrying all satellites, maneuvers from the injection orbit to the operational orbit of the first satellite and releases it. Subsequently, the OTV proceeds to deliver the remaining satellites to their respective operational orbits. Once all satellites are deployed, the OTV performs a de-orbit maneuver to enter a disposal orbit (with semi-major axis a_e) to avoid becoming space debris. In the disposal orbit, the OTV can gradually lower its altitude due to atmospheric drag, eventually burning up upon re-entry. This entire process is repeated over multiple launches until all satellites are deployed. It is assumed that the same type of launch vehicle and OTV are used for each launch, ensuring consistent payload capacity and maneuvering capability. Compared to direct or indirect L&D strategies, the OTV-based approach enables rapid deployment to multiple orbits without consuming the satellites' own propellant, offering the following advantages: 1) extended on-orbit lifespan for the satellites; 2) more efficient single-launch deployment process; and 3) broader range of orbital planes accessible in a single launch.

(3) L&D Optimization Problem

In a single launch, multiple satellites must be delivered to different orbital positions. Different combinations of satellites will result in varying fuel consumption for the OTV. Furthermore, even with the same satellite combination, the deployment sequence also affects fuel consumption. Given the OTV's limited fuel capacity, if the fuel consumed dur-

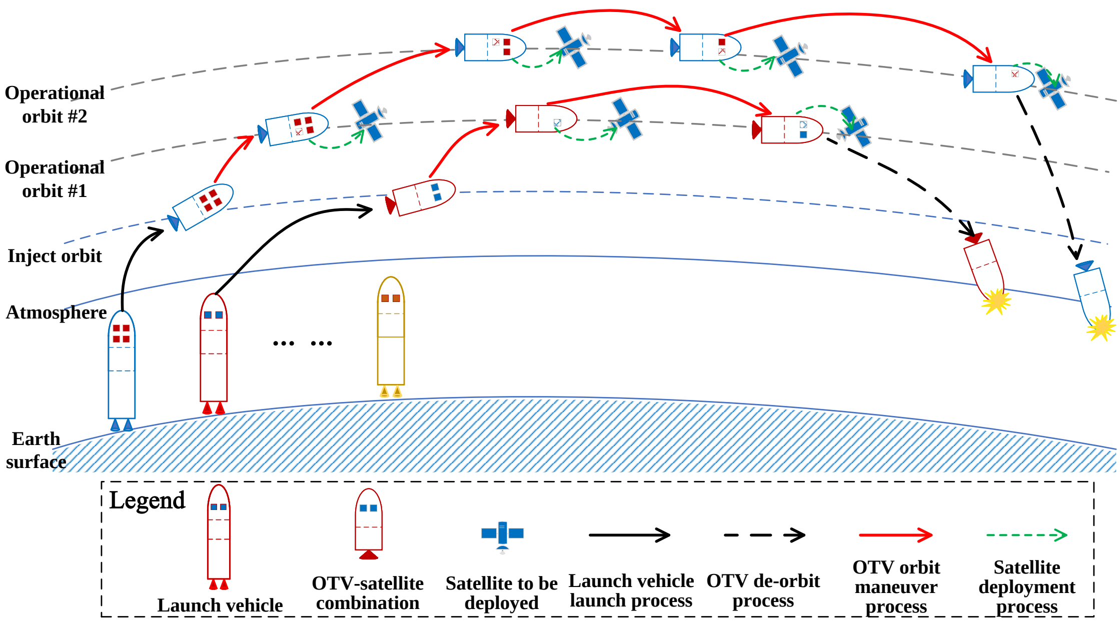


Figure 4.1: OTV-based launch and deployment process.

ing deployment exceeds this limit, the mission cannot be completed, necessitating additional launches. Therefore, different launch combinations and deployment sequences lead to variations in the launch count and fuel cost, which in turn affect the overall mission cost and the reliability of each launch. The L&D optimization problem aims to determine the satellite combination and deployment sequence for each launch to achieve the following objectives:

1) Minimizing the total number of launch missions. Given that launch vehicle procurement and operations represent a major financial component of constellation deployment—often surpassing individual satellite costs—reducing the frequency of launches is a direct and effective method for lowering overall construction expenditures.

2) Maximizing the average propellant margin of the OTV fleet. During orbital transfers, OTVs consume propellant to reach designated slots. Preserving a higher av-

erage fuel reserve beyond the nominal mission profile enhances the system's resilience, providing necessary capacity for contingency operations such as collision avoidance or injection error correction.

Simultaneously, the L&D process must satisfy the following resource and mission constraints:

1) OTV fuel capacity limit;
2) The launch vehicle payload capacity, constrained by factors such as fairing size;
3) The fuel consumption required for the OTV's controlled de-orbit after deployment.

4.3 Modeling of the Launch and Deployment Optimization Problem

This section first derives a rapid fuel consumption calculation model for an OTV performing continuous orbital maneuvers to deploy homogeneous satellites. Subsequently, drawing upon the capacitated Vehicle Routing Problem (CVRP), a 0-1 integer optimization model is formulated to describe the constellation L&D optimization problem. This model encompasses decision variables, optimization objectives, and constraints, culminating in a bi-objective optimization problem.

4.3.1 OTV Orbit Maneuver Fuel Calculation Model

The fuel consumption for an OTV to maneuver from orbital position s_i to s_j can be calculated using the Tsiolkovsky rocket equation [109]:

$$m_f^{ij} = m_i - m_i \exp\left(\frac{-\Delta V_{ij}}{g_0 I_{\text{sp}}}\right) \tag{4.1}$$

where ΔV_{ij} is the required velocity increment to travel from s_i to s_j. $g_0 = 9.80\ \mathrm{m/s^2}$ is the standard gravity at sea level, and I_{sp} is the specific impulse of the OTV's chemical thruster. m_i is the total mass of the OTV before the maneuver, which includes the OTV's dry mass, the mass of the remaining satellites, and the remaining fuel mass.

After each orbital maneuver and satellite release, the total mass of the OTV changes abruptly. Therefore, the total fuel consumption cannot be calculated by simply summing the velocity increments of multiple maneuvers. Instead, it must be calculated recursively. Assume an OTV carries n ($n \leq n_{\text{max}}$) homogeneous satellites, with a total initial mass of m_0, starting from an injection orbit to deploy the satellites into their respective operational orbits. In this process, the OTV performs n orbital maneuvers, releasing one satellite of mass m_s after each maneuver. If satellite releases are not considered, the total fuel consumption for n maneuvers is

$$m'_f = m_0 \left[1 - \exp\left(-\frac{\sum_{i=1}^{n} \Delta V_i}{g_0 I_{\text{sp}}}\right)\right] \tag{4.2}$$

where ΔV_i represents the velocity increment required for the OTV's i-th maneuver. The above equation calculates the total fuel consumption as if all satellites were carried through all n maneuvers. In reality, the i-th satellite released is no longer part of the subsequent $(n-i)$ maneuvers. Therefore, the excess fuel consumption accounted for the i-th released satellite in Eq. (4.2) is

$$m'_{fi} = m_s \left[1 - \exp\left(-\frac{\sum_{j=i+1}^{n} \Delta V_j}{g_0 I_{\text{sp}}}\right)\right] \tag{4.3}$$

Accordingly, the actual fuel consumption can be expressed as

$$
\begin{aligned}
m_f &= m_f' - \sum_{i=1}^{n-1} m_{fi}' \\
&= m_0 \left[1 - \exp\left(-\frac{\sum_{i=1}^{n} \Delta V_i}{g_0 I_{\text{sp}}}\right)\right] + \\
&\quad \sum_{i=1}^{n-1} m_s \exp\left(-\frac{\sum_{j=i+1}^{n} \Delta V_j}{g_0 I_{\text{sp}}}\right) - (n-1) m_s
\end{aligned}
\tag{4.4}
$$

This formula indicates that the actual fuel consumption after the OTV performs n orbital maneuvers is the initial fuel mass minus the sum of the remaining fuel mass after each maneuver. This model can be used to calculate the OTV's fuel consumption for transfers between different orbital points.

To calculate the velocity increment ΔV_{ij} for transfers between any two satellite operational orbits, this chapter employs the rapid analytical method proposed by [122]. This method is based on a two-impulse burn model. It accounts for the effects of J_2 perturbations on both the terminal position and the transfer orbit to determine the minimum required velocity increment. A constant transfer duration, denoted as T_m, is assumed for all inter-satellite maneuvers. For the specific cases of maneuvering from the injection orbit to the first satellite's orbit and from the last satellite's orbit to the disposal orbit, the velocity increment is calculated using the standard Hohmann transfer [109].

4.3.2 Bi-Objective Optimization Modeling for Constellation Launch and Deployment

The L&D optimization problem studied in this chapter primarily focuses on the OTV's deployment process and the pre-launch selection of satellite combinations. Therefore, the following assumptions are made regarding the launch site, launch vehicle characteristics, and launch procedures:

Assumption 4.1. *The launch site is fixed and can accommodate launches to various altitudes and inclinations required for LEO constellations.*

Assumption 4.2. *The same type of launch vehicle is used for all launches, ensuring a fixed payload capacity.*

Assumption 4.3. *The specified injection orbit altitude and inclination are within the launch vehicle's full payload capacity.*

Assumption 4.4. *Each launch vehicle represents a single launch and can carry only one OTV.*

Based on these assumptions, the OTV-based constellation L&D optimization problem can be modeled as a 0-1 nonlinear integer programming problem. The model, including decision variables, optimization objectives, and constraints, is described below.

4.3.2.1 Decision Variables

1. x_{ijk}: OTV deployment sequence indicator variable. x_{ijk} is defined as:

$$x_{ijk} = \begin{cases} 1, \text{ if OTV } k \text{ maneuvers from } s_i \text{ to } s_j \\ 0, \text{ otherwise} \end{cases} \tag{4.5}$$

where $i, j \in S'$ and $k \in K$. We define $S' = 0 \cup S = \{0, 1, ..., N\}$, where the "virtual satellite" 0 corresponds to the orbital position s_0 for both the injection and disposal orbits. $K = \{1, ..., N\}$ is the set of OTVs, where the OTV index represents the launch sequence. This binary decision variable describes the satellite combination for a single launch, the launch sequence of a particular combination, and the deployment sequence of satellites within that combination. When an OTV reaches an orbital point in the set S, the corresponding satellite is released and deployed. In the most conservative scenario, only one satellite is launched at a time, making the number of OTVs (i.e., the number of launches) equal to the number of satellites. Thus, the number of OTVs can be set equal to the number of satellites.

4.3.2.2 Optimization Objectives

$$f_1 = \min \sum_{k \in K} \sum_{j \in S} x_{0jk} \tag{4.6}$$

$$f_2 = \min \sum_{k \in K} \sum_{j \in S'} \sum_{i \in S'} m_f^{ijk} x_{ijk} \Bigg/ \sum_{k \in K} \sum_{j \in S} x_{0jk} \tag{4.7}$$

Eq. (4.6) minimizes the total number of required OTVs, which is equivalent to the number of launch missions. Since launch vehicle procurement and operations constitute the primary expense in constellation construction, minimizing frequency is the most direct approach to reducing overall L&D costs.

Eq. (4.7) seeks to minimize the average propellant consumption of the OTV fleet. Conserving fuel during nominal transfers increases the residual propellant margin, thereby enhancing system robustness by ensuring sufficient capacity for contingency operations, such as injection error correction and collision avoidance.

4.3.2.3 Constraints

(1) Orbital Transfer Path Constraints

$$\sum_{i \in S'} x_{ijk} = \sum_{i \in S'} x_{jik}, \quad \forall j \in S', k \in K \tag{4.8}$$

$$\sum_{k \in K} \sum_{i \in S'} x_{ijk} = 1, \quad \forall j \in S \tag{4.9}$$

$$\sum_{j \in S'} x_{0jk} = 1, \quad \forall k \in K \tag{4.10}$$

$$\sum_{i \in V, j \notin V} x_{ijk} \geq 2, \quad V \subset S, 2 \leq |V| \leq N-1 \tag{4.11}$$

Eq. (4.8) constrains the number of times an OTV enters and leaves each orbital position in S to be equal. Eq. (4.9) ensures that each orbital position in S is visited exactly once, and only one satellite is deployed at a time. Eq. (4.10), combined with (4.8), guarantees that each OTV starts from the injection orbit and eventually enters the disposal orbit. If $x_{00k} = 1$, it indicates that this launch is not performed. Eq. (4.11) is a sub-tour elimination constraint for the set S, preventing unreasonable sub-cycles in the deployment process.

(2) Orbital Transfer Fuel Consumption Constraints

The fuel consumption for each orbital transfer of an OTV must satisfy the following constraint:

$$m_f^{ijk} = m_{ik} \left[1 - \exp\left(\frac{-\Delta V_{ij}}{g_0 I_{\text{sp}}} \right) \right], \ \forall i \in S', j \in S, k \in K \tag{4.12}$$

where m_{ik} represents the remaining mass of OTV k after releasing satellite i. m_{ik} depends on the OTV's mass at the previous orbital point and the fuel consumed to reach the current position s_i. It is calculated as follows:

$$m_{ik} = \sum_{j \in S'} x_{jik} \left[\left(m_{jk} \exp \left(-\frac{\Delta V_{ji}}{g_0 I_{\text{sp}}} \right) \right) - m_s \right], \ \forall i \in S, k \in K \tag{4.13}$$

To obtain the remaining mass of the OTV at each orbital point, its initial mass in the injection orbit must first be determined. The initial mass of the k-th OTV is:

$$m_{0k} = m_{f0} + m_{\text{dry}} + m_s \sum_{i \in S'} \sum_{j \in S} x_{ijk}, \ \forall k \in K \tag{4.14}$$

where m_{f0} is the initial fuel mass of each OTV (assumed to be the same for all OTVs), and m_{dry} is the OTV's dry mass. Based on Eq.s (4.12)-(4.14), once the decision variables are determined, the fuel consumption for each orbital transfer of every OTV can be calculated iteratively.

(3) OTV De-orbit Fuel Consumption Constraint

$$m_f^{i0k} = m_{\text{dry}} \exp \left(\frac{\Delta V_{i0}}{g_0 I_{\text{sp}}} \right) - m_{\text{dry}}, \ \forall i \in S, k \in K \tag{4.15}$$

Eq. (4.15) specifies that the fuel required for each OTV's de-orbit maneuver is a constant, representing the minimum de-orbit fuel consumption. This minimum corresponds to the scenario where the OTV maneuvers to the disposal orbit after its fuel is depleted.

(4) Constraint on the Number of Satellites per Launch

Due to fairing size and satellite mass limitations, the number of satellites that can be carried in a single launch is limited. This is constrained as follows:

$$n_k = \sum_{i \in S'} \sum_{j \in S} x_{ijk} \leq n_{\max}, \ \forall k \in K \tag{4.16}$$

where n_k denotes the payload count of OTV k. The summation on the left calculates the actual carried satellites, consistent with the routing logic in Eqs. (4.8) and (4.9).

(5) OTV Fuel Constraint

$$m_{fk} = \sum_{i \in S'} \sum_{j \in S'} m_f^{ijk} x_{ijk} \leq m_{f0}, \ \forall k \in K \tag{4.17}$$

where m_{fk} is the total fuel consumption of OTV k. Eq. (4.17) ensures that the fuel consumption of each OTV does not exceed its initial onboard fuel capacity.

4.4 Solution Framework Combining the ϵ-Constraint Method and CNS-LSGA

This section outlines the solution architecture for the bi-objective L&D optimization task. The ϵ-constraint technique is primarily utilized to decompose the original problem into a sequence of single-objective subproblems. To tackle these subproblems effectively, the CNS-LSGA algorithm is developed.

4.4.1 Bi-Objective Optimization Framework Based on the ϵ-Constraint Method

The ϵ-constraint method, a scalarization technique widely used in multi-objective optimization, manages conflicting goals by optimizing a primary objective while bounding others within specified ϵ limits [121], [123]. By iteratively shifting ϵ with a step size Δ across the range defined by the ideal and Nadir points, the original problem is decomposed into a series of solved single-objective subproblems [121].

Given the discrete integer characteristic of the launch count f_1, it is designated as the constraint variable, while the continuous average fuel consumption f_2 acts as the primary optimization target. By defining a step size of $\Delta = 1$, the algorithm can exhaustively traverse the feasible domain of f_1, avoiding the granularity issues and computational overhead associated with decomposing the continuous f_2. The valid range for f_1 is strictly bounded by the Ideal point $f_1^I = \lceil N/n_{\text{max}} \rceil$ and the Nadir point $f_1^N = N$.

Accordingly, the subproblem $\boldsymbol{P}_1$ is defined as follows:

Objective function: Eq. (4.7),

Constraints: Eqs. (4.8)-(4.17), with the addition of the following ϵ-constraint:

$$\sum_{k \in K} \sum_{j \in S} x_{0jk} \leq \epsilon \tag{4.18}$$

where $\epsilon \in [f_1^I, f_1^N]$.

For each subproblem $\boldsymbol{P}_1$ with a different value of ϵ, the proposed CNS-LSGA algorithm is used for solving. Due to the coupling between the fuel constraint and the maximum number of satellites per launch constraint, the solution corresponding to the ideal point f_1^I may not be feasible. Therefore, a feasibility check is required for all solutions, and infeasible ones must be discarded. The ϵ-constraint method based solution framework is outlined in Algorithm 4-1.

Algorithm 4-1 ϵ-constraint method based L&D solution framework[124]

Require: Satellite set C; Satellite orbit position set S; Satellite mass m_s; Maximum satellite carrying number of the launch vehicle n_{max}; OTV dry mass m_{dry}; OTV initial fuel mass m_{f0}; OTV specific impulse I_{sp}; Disposal orbit semi-major axis a_e; Injection orbit semi-major axis a_0.

Ensure: The non-dominated objective set $Pareto$, and the solution set $\boldsymbol{X}$.

1: Initialize $Pareto = \varnothing$, $\boldsymbol{X} = \varnothing$;
2: Set $f_1^N = N$, $f_1^I = \lceil N/n_{\text{max}} \rceil$;
3: $f_2^I \leftarrow$ Calculate the fuel cost of deploying one satellite and de-orbit;

$Pareto \leftarrow Add([f_1^N, f_2^I])$;
Add the L&D plan of one launch vehicle launch one satellite to $\boldsymbol{X}$;
Set $\epsilon = f_1^N$, $\Delta = 1$;
for i = 1 to $(f_1^N - f_1^I)$ **do**
  Set L&D plan $X_i = \varnothing$;
  $\epsilon = \epsilon - \Delta$;
  $\{f_2^i, X_i\} \leftarrow$ Solve problem $\boldsymbol{P}_1$ with ϵ via the CNS-LSGA (described in Section 4.4.2);
  if X_i satisfies (4.8)-(4.17) **then**
    Get the value of f_1^i by X_i;
    $\boldsymbol{X} \leftarrow Add(X_i)$;
    $Pareto \leftarrow Add([f_1^i, f_2^i])$;
  end if
end for
Remove the dominated solutions in $Pareto$ and $\boldsymbol{X}$;

4.4.2 CNS-LSGA Algorithm Framework

4.4.2.1 Decision Variable Encoding

When using a Genetic Algorithm to solve an optimization problem, it is crucial to first define an individual encoding scheme suitable for genetic operators. This scheme transforms the problem's decision variables into a format that genetic operators can manipulate. The representation of decision variables significantly impacts the efficiency of solving combinatorial optimization problems, and a well-designed encoding method can markedly improve the optimization performance for the L&D problem.

For problem $\boldsymbol{P}_1$, a splitter-based encoding strategy is adopted (Fig. 4.2). An individual is defined as a permutation of length $(N+p-1)$, comprising satellite indices $\{1, ..., N\}$ and splitters $\{(N+1), ..., (N+p-1)\}$. Subsequences bounded by these splitters correspond to the deployment order for distinct launches. Consequently, the complete L&D plan is extracted by segmenting the permutation, with the splitter count regulating the maximum

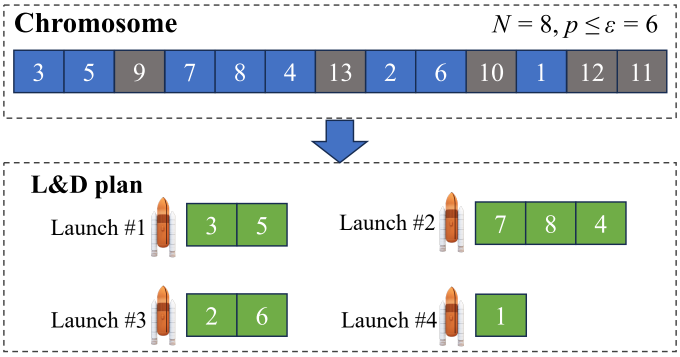


Figure 4.2: Splitter-based solution representation method.

number of launches.

Setting the splitter count to $(\epsilon - 1)$ naturally enforces the launch number limit in Eq. (4.18). However, since this encoding does not guarantee adherence to payload capacity and fuel constraints, penalty terms are integrated into the objective function to construct the final fitness metric:

$$F_{\text{fit}} = f_2 + Pen_1 \max\{\sum_{k \in K}(n_k - n_{\max}), 0\} + Pen_2 \max\{\sum_{k \in K}(m_{fk} - m_{f0}), 0\} \quad (4.19)$$

where Pen_1 and Pen_2 are the penalty coefficients.

4.4.2.2 Cluster and Nearest Sorting Initialization Mechanism

To efficiently solve problem $\boldsymbol{P}_1$ using the splitter-based representation, a specialized initialization strategy is employed to accelerate GA convergence. Since deploying proximate satellites significantly reduces fuel consumption, this method generates high-quality initial solutions by clustering satellites based on orbital parameters. Specifically, the constellation is partitioned into subsets, where each subset represents a single launch combination, followed by the generation of internal deployment sequences.

The K-means clustering method [125] is applied to group the satellites. The K-means method requires a predefined number of clusters and assigns satellites to the cluster with the nearest centroid. Different orbital parameters of the satellites are selected as the primary metrics for clustering, including semi-major axis, inclination, RAAN, and true anomaly (or argument of latitude). Differences in RAAN and inclination primarily reflect different orbital planes. The difference in the argument of latitude on the orbit is jointly determined by the inclination, RAAN, and true anomaly. Furthermore, the velocity increment for phasing maneuvers is also affected by the number of phasing orbits, k_{phase}. If the standard Euclidean distance is used directly as the clustering metric, it implicitly assumes that the differences in all orbital elements have the same impact on the orbital transfer cost. However, the influence of different orbital elements on the velocity increment for orbital maneuvers is not uniform. To address this, this chapter introduces a weighted Manhattan distance metric, which more accurately reflects the actual impact on maneuver requirements by adjusting the weights of each orbital element. The specific weighted Manhattan distance is defined as:

$$d = \lambda_1 \left| a_i - a_j \right| + \lambda_2 \left| i_i - i_j \right| + \lambda_3 \left| \Omega_i - \Omega_j \right| + \lambda_4 \left| u_i - u_j \right|, \; i, j \in C \tag{4.20}$$

where $\{\lambda_1, \lambda_2, \lambda_3, \lambda_4\} \in [0, 1)$ are weight coefficients used to adjust the relative importance of each orbital element in the distance metric. For most constellations, to maintain a stable configuration, the inclination and semi-major axis are usually kept constant, so λ_2 and λ_1 can be set to 0. λ_3 and λ_4 are adjusted according to the actual orbital parameters of the satellites to reflect the impact of RAAN and true anomaly on orbital maneuvers. In a few multi-layered or SSO constellations where satellites may have different inclinations and semi-major axes, the values of λ_1 and λ_2 can be adjusted accordingly.

For the clustered satellite subsets, a nearest-neighbor sorting method is proposed to systematically generate launch combinations and deployment sequences, while simultaneously satisfying the constraints on the maximum number of satellites per launch (4.16) and the OTV fuel limit (4.17). This method consists of two main stages:

(1) Nearest-neighbor sorting. A random satellite within the subset serves as the starting node. The algorithm iteratively selects the next satellite based on the minimum velocity increment required from the current node, continuing until the entire subset is traversed.

(2) Greedy sequence partitioning. Following the sorted sequence, $n_{\max}$ satellites are successively selected to form a candidate launch combination. The OTV fuel consumption is then calculated to check if it satisfies constraint (4.17). If not, satellites are removed from the end of the sequence one by one until the fuel constraint is met. The removed satellites are returned to the sequence for subsequent grouping.

The sequences generated above may violate the launch number constraint (4.18). To address this, a nearest-neighbor reassignment mechanism is introduced: first, all launch combinations are sorted in ascending order by the number of satellites. The redundant combinations with the fewest satellites are identified, and the satellites in these combinations are reassigned to other launch combinations that are closest in orbital position and have not exceeded the satellite limit ϵ. This mechanism can adjust some launch combinations that violate the OTV fuel constraint, ensuring that the final result satisfies the launch number constraint and conforms to the individual encoding scheme. Finally, all grouped sequences and splitters together form an individual in the population.

Algorithm 4-2 details the Cluster and Nearest Sorting (CNS) initialization procedure. To ensure population diversity, three randomization strategies are integrated: 1) the cluster count varies randomly for each individual; 2) deployment sequences and splitters are randomly positioned; and 3) only a specific ratio, $P_{cluster}$, of the population is initialized via CNS, with the remainder generated randomly.

Algorithm 4-2 Cluster and nearest sorting initial solution generation[124]

Require: Satellite set C; Satellite orbit position set S; maximum launch number ϵ; The proportion of individuals generated by clustering methods $P_{cluster}$; Population size N_{pop}; Distance metric coefficient λ;maximum satellite carrying number of the launch vehicle $n_{\max}$.

Ensure: The initial population Pop.

1: Set $N_{pop}^{cluster} = \lceil N_{pop} \times P_{cluster} \rceil$, $N_{pop}^{rand} = N_{pop} - N_{pop}^{cluster}$;
2: $\Delta VSet \leftarrow$ Calculate velocity increment between satellites in C by method in [122];
3: **for** i = 1 to $N_{pop}^{cluster}$ **do**
4:   $\boldsymbol{R} \leftarrow \varnothing$;
5:   *// Satellite cluster via K-means (line 6-7)*
6:   Cluster number: $N_{cluster} \leftarrow$ Random integer between 2 to ϵ;
7:   Cluster result: $Idx \leftarrow$ *K-means cluster*(S, $N_{cluster}$, λ);
8:   **for** j = 1 to $N_{cluster}$ **do**
9:     *// Nearest neighbor satellite sorting (line 10-16)*
10:     $ClusterSet \leftarrow Find(Idx == j)$;
11:     Start satellite: $c \leftarrow$ Random element in $ClusterSet$;
12:     Sorted satellite sequence: $Order \leftarrow c$;
13:     **for** k = 2 to $|ClusterSet|$ **do**
14:       $c' \leftarrow Find(\Delta V(c, c') == \min(\Delta VSet\{c, ClusterSet \backslash Order\}))$;
15:       $Order \leftarrow Add(c')$, $c \leftarrow c'$;
16:     **end for**
17:     *// Greedy sequence partitioning (line 18-28)*
18:     **while** $Order \neq \varnothing$ **do**
19:       $idx \leftarrow \min\{n_{\max}, |Order|\}$;
20:       $R' \leftarrow Order(1 : idx)$;
21:       $m_f(R') \leftarrow$ Calculate fuel cost by Eq. (4.4);
22:       **while** $m_f(R') > m_{f0}$ **do**
23:         $idx = idx - 1$; $R' \leftarrow Order(1 : idx)$;
24:         $m_f(R') \leftarrow$ Calculate fuel cost by Eq. (4.4);
25:       **end while**
26:       $\boldsymbol{R} \leftarrow Add(R')$, $Order(1 : idx) = \varnothing$;
27:     **end while**
28:   **end for**
29:   *// Supplementary nearest reassignment (line 30-40)*
30:   **if** $|\boldsymbol{R}| > \epsilon$ **then**
31:     $\boldsymbol{R} \leftarrow$ Descending sorting R in $\boldsymbol{R}$ according the satellite number in R;
32:     $Order \leftarrow$ Combine deployment sequences (from $\boldsymbol{R}\{\epsilon+1\}$ to $\boldsymbol{R}\{end\}$) into one sequence;

33: $\boldsymbol{R} \leftarrow$ Remove $\boldsymbol{R}\{\epsilon+1\}$ to $\boldsymbol{R}\{end\}$;
34: **for** j = 1 to $|Order|$ **do**
35: $\boldsymbol{R}' \leftarrow Find(|R| < n_{\max}, R \in \boldsymbol{R})$;
36: Find $R' \in \boldsymbol{R}$ with minimum average ΔV from $Order(j)$ to R';
37: Insert $Order(j)$ to a random position in R';
38: update R' in $\boldsymbol{R}$;
39: **end for**
40: **end if**
41: *// Combine deployment sequence to initial individual (line 42-59)*
42: $Splitter \leftarrow$Generate random sequence from $(N+1)$ to $(N+\epsilon-1)$;
43: **for** j=1 to $|\boldsymbol{R}|-1$ **do**
44: $Ind \leftarrow Add([\boldsymbol{R}\{j\}, Splitter(j)])$;
45: $Splitter(j) = \varnothing$
46: **end for**
47: $Ind \leftarrow Add([\boldsymbol{R}\{end\}, Splitter])$
48: $Pop \leftarrow Add(Ind)$;
49: **end for**
50: *// Random individual generation (line 51-53)*
51: **for** $i = (N_{pop}^{cluster}+1)$ to N_{pop} **do**
52: Initialize Ind as random sequence from 1 to $(N+\epsilon-1)$;
53: $Pop \leftarrow Add(Ind)$;
54: **end for**

4.4.2.3 Genetic Operators

To select individuals for the next generation, a tournament selection strategy [126] is employed. This method randomly picks a subset of candidates and retains the one with the highest fitness, thereby favoring superior solutions in the evolutionary process. For reproduction, the Order Crossover (OX1) operator [127], designed for permutation encoding, is applied with probability P_c. It inherits a random subsequence from one parent while filling the remaining slots based on the order of the second parent. Finally, mutation is conducted via the Swap mutation method [128] with probability P_m, where genes at two randomly chosen positions are exchanged.

4.4.2.4 Local Search Operators

Basic GA may easily become trapped in local optima in large-scale problems due to a lack of efficient local exploitation mechanisms [71]. To address this limitation, this chapter introduces two local search mechanisms into the basic GA process, named *Two-sequence swap* and *One-sequence re-order*.

In the *Two-sequence swap* operator, two deployment sequences are randomly selected from within an individual. Then, two satellite positions are randomly chosen in one sequence, and their corresponding satellite IDs are swapped with two randomly chosen satellites from the other sequence. This operator attempts to find better solutions by randomly exchanging parts of deployment sequences between different OTVs.

The *One-sequence re-order* operator randomly selects a single deployment sequence within an individual and reorders it based on a greedy strategy. Specifically, a satellite is randomly chosen as the starting point. Each subsequent satellite is selected as the one requiring the minimum velocity increment from the last satellite in the current sequence. This process continues until all satellites in the sequence have been reordered.

To improve computational efficiency, local search is not applied to the entire population, as this would be computationally expensive. Instead, it is only performed on the top P_{LS} proportion of individuals, ranked by fitness. A selected individual will undergo one of the two local search operators, chosen randomly. If a better solution is found, the new individual is added to the population. To maintain a constant population size, the individuals with the lowest fitness are removed after each iteration. The detailed process of the local search is described in Algorithm 4-3.

Algorithm 4-3 Local search[124]

Require: Current population Pop; Satellite orbit position set S; OTV initial fuel m_{f0}.
Ensure: The new population $newPop$.

Set $newPop = Pop$;
for i = 1 to $\lceil P_{LS} \times |Pop| \rceil$ **do**
    Deployment sequence set: $\boldsymbol{R} \leftarrow decode(Pop\{i\})$;
    if $Rand < 0.5$ **then**
        $\{R_a, R_b\}$ $\leftarrow$Randomly select two deployment sequences from $\boldsymbol{R}$;
        if $|R_a| > 2$ && $|R_b| > 2$ **then**
            $\{g_{a1}, g_{a2}\}$,$\{g_{b1}, g_{b2}\}$ $\leftarrow$Randomly select two satellite in R_a and R_b, respectively;
            Exchange values between $\{g_{a1}, g_{a2}\} \in R_a$ and $\{g_{b1}, g_{b2}\} \in R_b$;
            $[m_{fa}, m_{fb}]$ $\leftarrow$Calculate the total fuel cost of R_a and R_b by Eq. (4.4);
            if $m_{fa} \leq m_{f0}$&&$m_{fb} \leq m_{f0}$ **then**
                $Ind = Pop\{i\}$;
                Update Ind based on R_a and R_b;
                $newPop \leftarrow Add(Ind)$;
                Update the fitness value of Ind;
            end if
        end if
    else
        R_a $\leftarrow$Randomly select one deployment sequence from R;
        Set $R_b = \varnothing$, $R'_a = R_a$;
        Randomly select one satellite from R'_a and add it to R_b;
        while $|R'_a| > 0$ **do**
            s_b $\leftarrow$Get the orbit position of the last satellite in R_b;
            Calculate the velocity increment between s_b and satellites in R'_a by method in [122];
            g'_a $\leftarrow$Find the satellite in R'_a with minimum velocity increment from s_b;
            Add g'_a to R_b and remove it from R'_a;
        end while
        $m_{fb} \leftarrow$ Calculate the total fuel cost of R_b by Eq. (4.4);
        if $m_{fb} \leq m_{f0}$ **then**
            $Ind = Pop\{i\}$;
            Update Ind based on R_b;
            $newPop \leftarrow Add(Ind)$;
            Update the fitness value of Ind;
        end if
    end if
end for
Rank the individuals in $newPop$ in descending order based on the fitness value;
Delete individuals ranked $(|newPop| - |Pop| + 1)$ to $|Pop|$ from $newPop$;

4.4.2.5 Overall Process of CNS-LSGA

The overall process of the CNS-LSGA algorithm is illustrated in Fig. 4.3. First, based on the given mission parameters, initial solutions are generated using Algorithm 4-2. After fitness evaluation, three genetic operators—selection, crossover, and mutation—are sequentially executed to produce a new generation. Subsequently, the new population undergoes local search via Algorithm 4-3 to further improve solution quality. This iterative process of genetic operations and local search continues until the termination condition is met, ultimately outputting the optimal launch and deployment (L&D) plan.

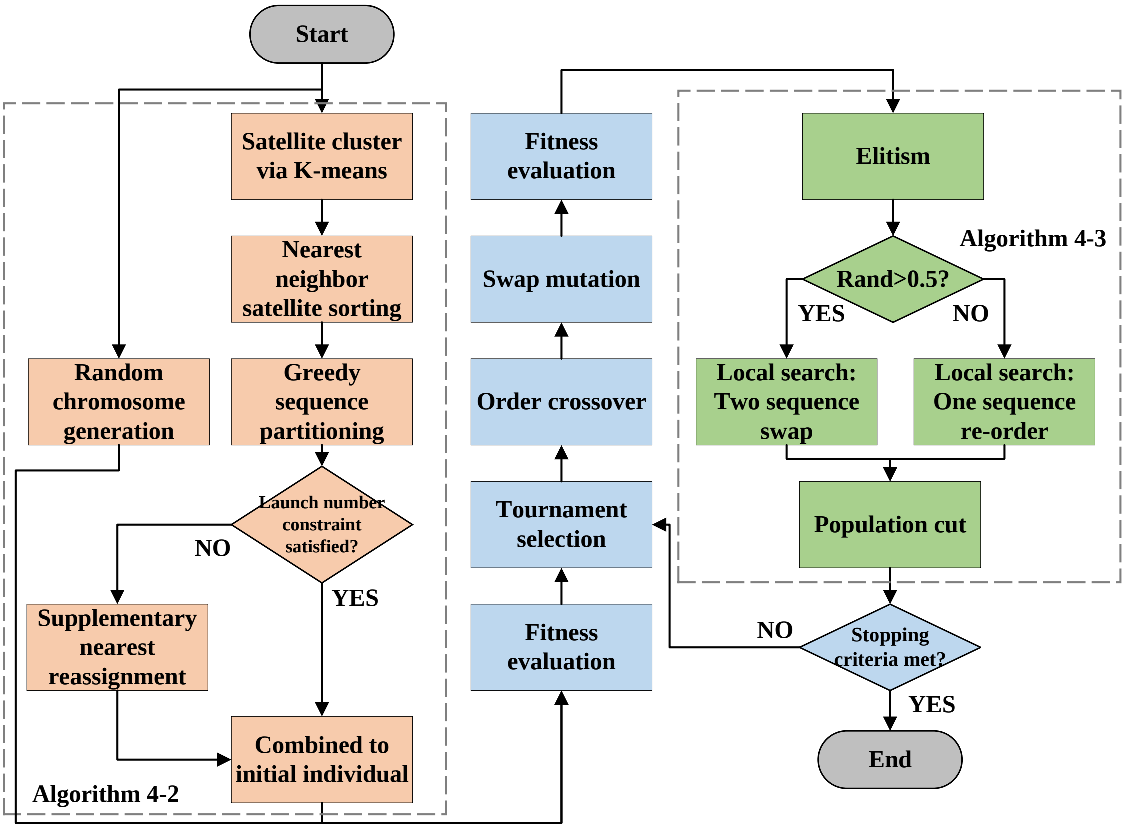


Figure 4.3: Overall process of CNS-LSGA.

4.5 Simulation of Case Study

This section validates the proposed model and solution framework by generating L&D strategies for different LEO constellations. Initially, the performance of the CNS-LSGA algorithm is evaluated through orthogonal, ablation, and comparative experiments on regional coverage constellations of varying scales. Subsequently, the proposed framework is benchmarked against established population-based multi-objective algorithms across three distinct scenarios: single-RGT and multi-RGT regional coverage constellations, as well as multi-shell Walker constellations[1]. The impact of the orbital maneuver duration parameter T_m is also analyzed. Simulations were performed on a laptop equipped with an Intel i7-12700H CPU and 16 GB RAM using MATLAB R2023b. Key parameters for the OTV, satellites, and algorithm settings are listed in Table 4.1.

Table 4.1: Parameters of the OTV, satellite and CNS-LSGA.

Parameter	Regional coverage single-RGT constellation case	Multi-shell Walker constellation case	Regional coverage multi-RGT constellation case
Launch vehicle inject orbit altitude	300 km		1000 km
OTV de-orbit altitude		200 km	
OTV dry mass m_{dry}		500 kg	
OTV initial fuel mass m_{f0}		2000 kg	
OTV thruster specific impulse I_{sp}		326.53 s	
Maximum number of satellites per launch $n_{\max}$	5	8	8
OTV single orbit maneuver time T_m	$5 \times T_{orbit}$	10 hours	$10 \times T_{orbit}$
Satellite mass m_s	800 kg	300 kg	400 kg
Proportion of initial solutions generated by CNS mechanism $P_{cluster}$		25%	
Penalty coefficients in fitness function $[Pen_1, Pen_2]$		[1000,100]	

T_{orbit} is the minimum orbital period of the satellite in the test constellation.

[1]Test constellation parameters are available at: https://github.com/HITPengHan/launch-and-deployment

4.5.1 Results on Regional Coverage Single-RGT LEO Constellation

Using the approach detailed in Chapter 3, three regional single-fold continuous coverage constellations based on RGT orbits were generated targeting mainland China. By varying the minimum elevation angle from $10°$ to $30°$, constellations of different scales were obtained, with specific parameters listed in Table 4.2 and satellite distributions visualized in the $\Omega - M$ frame in Fig. 4.4. Since OTV phasing costs in these RGT-based constellations are primarily driven by RAAN differences rather than mean anomaly, Eq. (4.20) coefficients are configured as $\lambda_1 = \lambda_2 = 0$, $\lambda_3 = 0.9$, and $\lambda_4 = 0.1$.

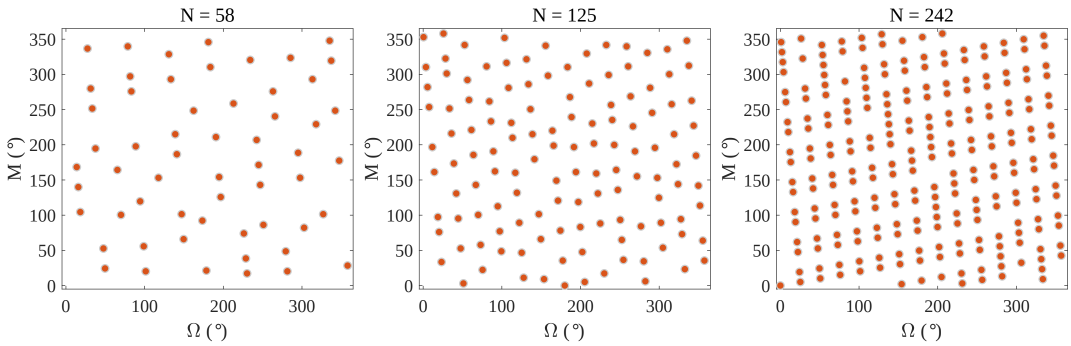


Figure 4.4: Satellite distribution of different single-RGT constellations in the $\Omega - M$ frame.

Table 4.2: Single RGT-based regional coverage constellations parameters.

Constellation number	Minimum elevation angle/°	Semimajor axis/km	Inclination/°	RAAN range/°	Number of satellites
1	10			[13, 358]	58
2	20	7203.7	45	[0, 358]	125
3	30			[1, 357]	242

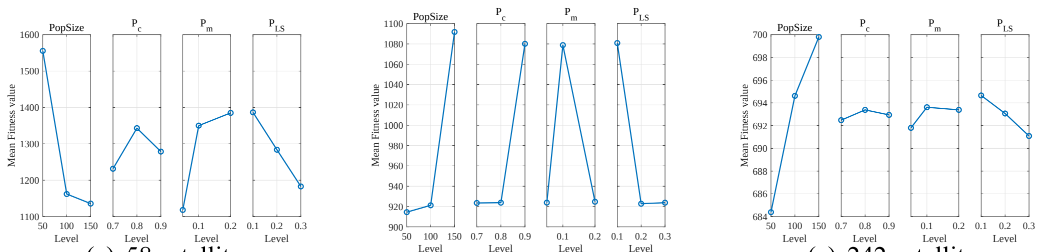
To validate the single-objective CNS-LSGA, nine $\boldsymbol{P}_1$ sub-problem instances are formulated by applying three distinct launch count constraints across each regional coverage LEO constellation scenario, with iteration limits set as follows:

- 58 satellites case: $\epsilon = \{15, 20, 25\}$, iteration number: 150000,
- 125 satellites case: $\epsilon = \{30, 40, 50\}$, iteration number: 200000,
- 242 satellites case: $\epsilon = \{50, 70, 90\}$, iteration number: 250000.

4.5.1.1 Orthogonal Experiment for CNS-LSGA

To identify optimal settings for the single-objective CNS-LSGA, a Taguchi orthogonal experiment [129] is performed. Four parameters, including population size (N_{pop}), crossover probability (P_c), mutation probability (P_m), and local search proportion (P_{LS}), are evaluated at three specific levels: $N_{pop} = \{50, 100, 150\}$, $P_c = \{0.7, 0.8, 0.9\}$, $P_m = \{0.05, 0.1, 0.2\}$, and $P_{LS} = \{0.1, 0.2, 0.3\}$. The experiments are conducted across the nine previously defined test cases. Each parameter combination undergoes 10 independent runs, using the average result as the response variable. Based on the analysis of the main effects plotted in Fig. 4.5, the optimal parameter configurations for constellations of varying scales are determined as follows:

- 58 satellites case: $N_{pop} = 150$, $P_c = 0.7$, $P_m = 0.05$, $P_{LS} = 0.3$,
- 125 satellites case: $N_{pop} = 50$, $P_c = 0.7$, $P_m = 0.05$, $P_{LS} = 0.3$,
- 242 satellites case: $N_{pop} = 50$, $P_c = 0.7$, $P_m = 0.05$, $P_{LS} = 0.3$.



(a) 58 satellites (b) 125 satellites (c) 242 satellites

Figure 4.5: Orthogonal experiment results of CNS-LSGA for tuning the best parameters.

4.5.1.2 Ablation Experiment for CNS-LSGA

To assess the effectiveness of the proposed components in CNS-LSGA, specifically the CNS initialization and the Local Search (LS) mechanisms, ablation studies were conducted. Four variants were compared using identical parameters derived from the earlier orthogonal experiments: (1) GA (standard genetic algorithm with random initialization); (2) CNS-GA (GA with CNS only); (3) LSGA (GA with LS only); and (4) the proposed CNS-LSGA. These algorithms were tested across the nine define sub-problems, with ten independent runs for each. The best average convergence curves are presented in Fig. 4.6.

The results in Fig. 4.6 demonstrate that CNS-LSGA consistently secures feasible solutions and outperforms the baselines across all scenarios. The benefits of specific components vary by problem difficulty. When launch constraints are tight (top row of figures), the LS operator plays a crucial role, allowing CNS-LSGA to significantly surpass CNS-GA. Conversely, as the satellite count increases, the CNS initialization becomes dominant. In the 242-satellite case, standard GA and LSGA failed to find feasible solutions, whereas the initial populations of CNS-LSGA and CNS-GA already outperformed the converged results of the other methods.

These findings confirm that integrating CNS initialization with LS operators successfully balances global exploration and local exploitation. While baseline algorithms struggled with constraint violations as complexity increased, CNS-LSGA maintained robustness, proving that both mechanisms are essential for high-quality convergence in large-scale instances.

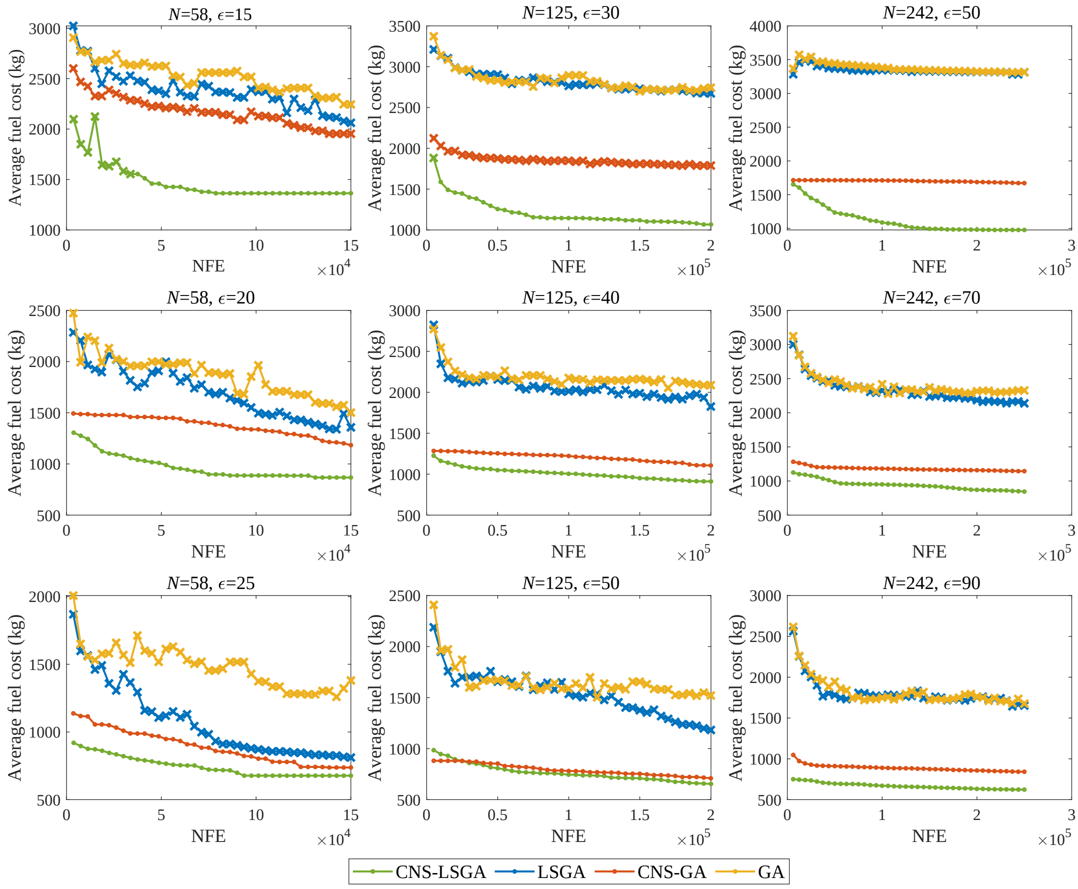


Figure 4.6: CNS-LSGA ablation results across sub-problems. Crosses indicate theoretical values involving constraint violations, while dots denote feasible iteration outcomes.

4.5.1.3 Comparative Experiment for CNS-LSGA

To validate the performance of the proposed CNS-LSGA, this section compares it against four single-objective metaheuristics commonly applied to MSSRP variants: ACO [104], IMODE [130], PSO [70], and LNS-AGA [131]. These algorithms were implemented using PlatEMO [120] and tested across the nine previously defined scenarios. To ensure statistical reliability, ten independent runs were conducted for each case, with population sizes and iteration limits matching those of CNS-LSGA. The resulting solution distributions and feasibility rates are summarized in Fig. 4.7.

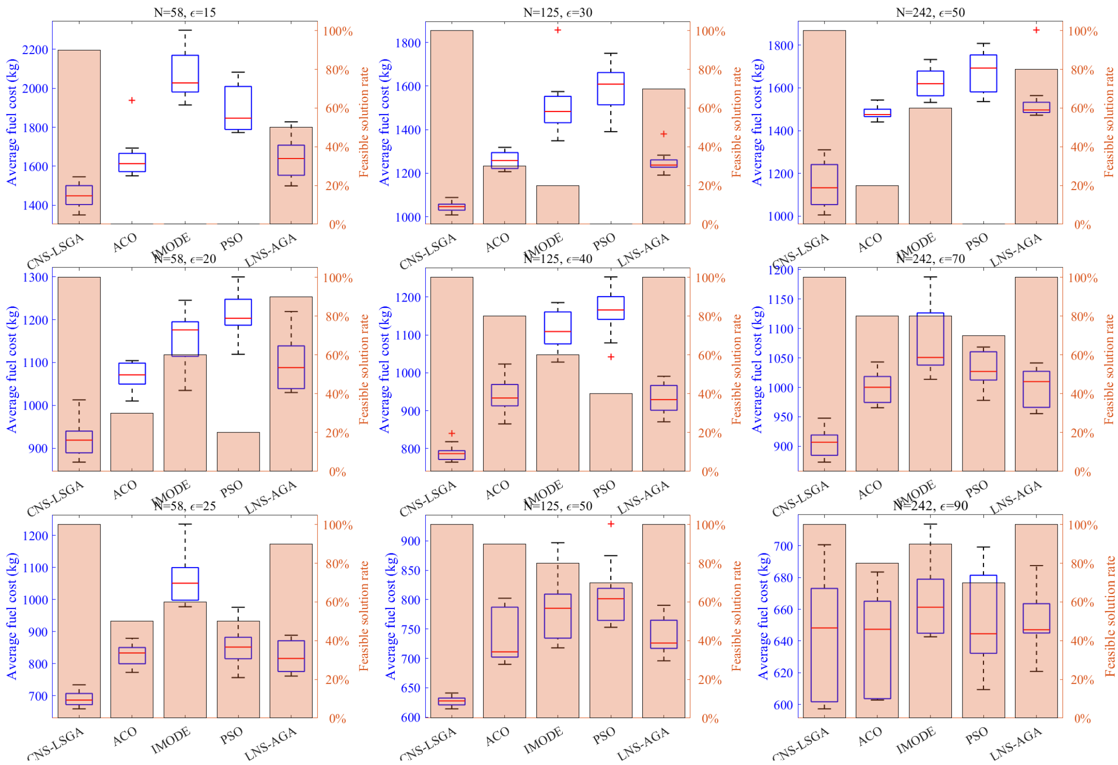


Figure 4.7: Comparison of algorithmic performance across distinct test instances.

As illustrated in Fig. 4.7, CNS-LSGA achieves superior mean objective values in eight of the test cases and demonstrates higher stability, evidenced by the compact boxplots. Crucially, the proposed algorithm consistently generates feasible solutions regardless of problem scale or launch constraints. In contrast, benchmarks such as ACO, IMODE, and PSO fail to locate feasible solutions under strict launch limitations (e.g., the $N = 58, \epsilon = 15$ scenario). While LNS-AGA exhibits comparable feasibility robustness, its optimization performance consistently lags behind CNS-LSGA. These results confirm the proposed algorithm's ability to effectively balance fuel efficiency with strict constraints, thereby significantly enhancing the quality of Pareto fronts generated within the ϵ-constraint framework.

4.5.1.4 Bi-Objective Optimization Result Comparison with Other MOEAs

The proposed ϵ-constraint framework with CNS-LSGA was evaluated across three constellation scales using 14-thread parallel computing. Benchmarking was conducted against established population-based multi-objective algorithms: NSGA-II [132], MOEA/D [133], and MSCMO [134], configured with a population size of 200 and 500,000 iterations. To ensure practical viability, only Pareto front solutions within specific launch count thresholds (30, 70, and 120, respectively) were retained. Fig. 4.8 illustrates the resulting Pareto fronts. Key performance metrics, including CPU time, non-dominated solution count, varying objective bounds, and the Hypervolume (HV) value [135] are detailed in Table 4.3.

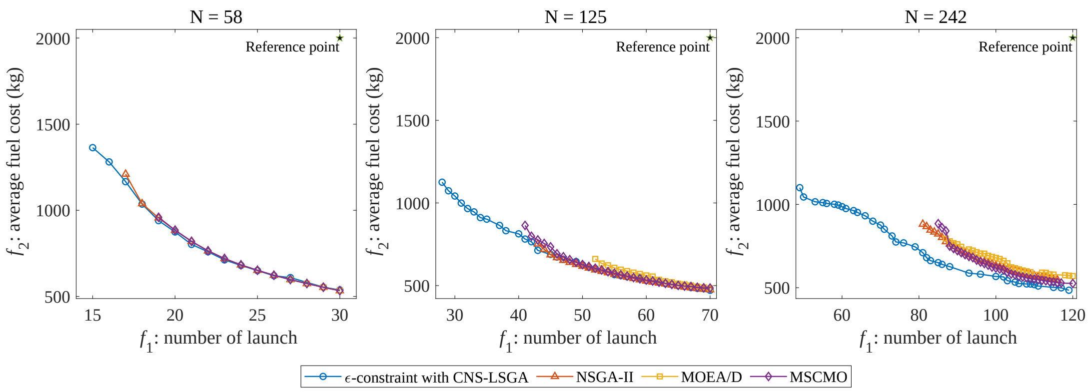


Figure 4.8: Pareto fronts generated by four algorithms across three constellation scales.

As shown in Fig. 4.8, the proposed framework generates effective Pareto fronts across all scenarios, confirming its robustness in handling multi-scale problems. The superior HV values in Table 4.3 reveal the framework's capability to produce more diverse and optimal solutions, exceeding comparison algorithms by 7.7% to 225.8% in the HV metric. Furthermore, the method significantly optimizes launch frequency, achieving minimum launch counts 12-44% lower than the baselines. This expanded range of superior solutions offers decision-makers greater flexibility. Although NSGA-II exhibits a negligible advantage in minimum fuel consumption for the 58-satellite case, this does not diminish the overall dominance of the proposed method on the Pareto front. The substantial improvements in diversity, optimality, and practical solution quantity validate the framework's effectiveness for complex L&D optimization.

Table 4.3: L&D result comparison of the four algorithms under three regional coverage constellations test case.

Constellations scale	Algorithm	CPU time	HV	Number of non-dominated solutions	$f_{1\,\min}$	$f_{1\,\max}$	$f_{2\,\min}$	$f_{2\,\max}$
58 satellites	ϵ-constraint with CNS-LSGA	**77.31 s**	**0.3489**	**15**	**15**	30	537.5	**1363.7**
	NSGA-II	148.52 s	0.3240	14	17	30	**533.2**	1209.9
	MOEA/D	254.65 s	0.1360	4	27	30	534.3	597.8
	MSCMO	175.55 s	0.2943	12	19	30	533.2	959.3
125 satellites	ϵ-constraint with CNS-LSGA	465.76 s	**0.4435**	**37**	**28**	70	**471.7**	**1125.6**
	NSGA-II	**278.17 s**	0.3304	28	43	70	475.2	752.4
	MOEA/D	454.58 s	0.2453	19	52	70	483.8	661.5
	MSCMO	306.72 s	0.3443	30	41	70	485.3	864.2
242 satellites	ϵ-constraint with CNS-LSGA	806.76 s	**0.4304**	**39**	**49**	119	**484.9**	**1100.4**
	NSGA-II	**508.29 s**	0.2751	36	81	116	546.8	881.4
	MOEA/D	624.21 s	0.2427	29	87	**120**	569.9	794.9
	MSCMO	529.98 s	0.2586	34	85	**120**	525.5	884.4

Reference points for calculating HV: [30,2000], [70,2000], [120,2000].

Regarding computational efficiency, parallel implementation allows the framework to achieve reasonable execution times across various constellation scales. While it outperforms population-based multi-objective algorithms for smaller problems (e.g., 58 satellites), its execution time exceeds these methods as the scale increases (see Table 4.3). This increased cost arises from two factors: larger scales yield more Pareto front solutions, necessitating more sub-problems, and the complexity of individual sub-problems grows. Conversely, population-based algorithms are primarily influenced by problem scale rather than solution quantity, maintaining relatively stable times. Nevertheless, since L&D planning is strategic pre-planning rather than a real-time operation, solution optimality and diversity take precedence over speed. The proposed framework prioritizes these aspects, offering superior decision options within timelines acceptable for practical mission planning.

4.5.1.5 L&D Plan Result Visualization

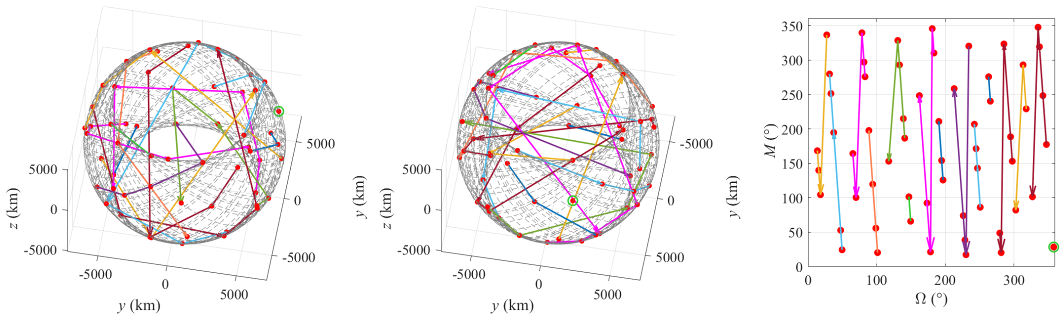
To validate the rationality of the generated strategies, representative Pareto-optimal solutions for the 58-satellite case (Constellation 1) are visualized in Fig. 4.9. It is evident that the grouping of launch combinations strictly corresponds to the optimized objective value f_1. A clear clustering pattern emerges where satellites with adjacent RAAN values are assigned to the same launch vehicle, with the deployment sequence generally following the RAAN order.

This phenomenon stems from the specific configuration of the RGT constellation, characterized by widely dispersed orbital planes containing single satellites. Given that plane-change maneuvers constitute the primary fuel expenditure, the algorithm prioritizes grouping satellites within similar orbital planes to maximize payload utilization and minimize overall energy costs.

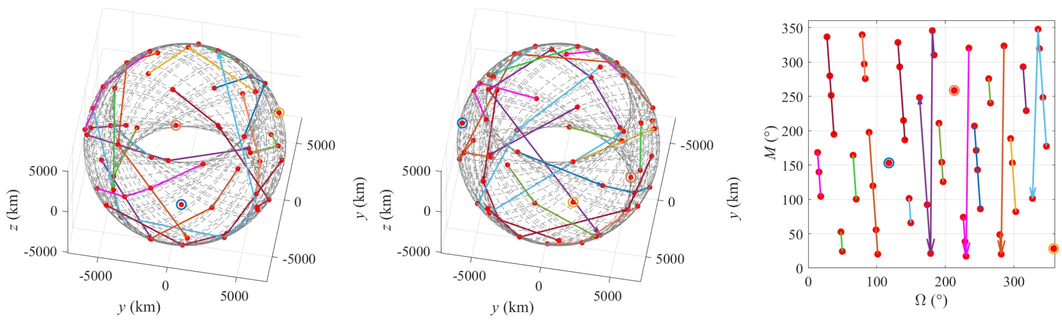
4.5.1.6 Sensitivity Analysis of Orbital Maneuver Duration

The orbital maneuver duration T_m directly dictates fuel consumption and consequently shifts the distribution of Pareto-optimal solutions. To quantify this impact, the 58-satellite constellation case was tested with T_m values ranging from $3 \times T_{orbit}$ to $30 \times T_{orbit}$. The resulting Pareto fronts are plotted in Fig. 4.10.

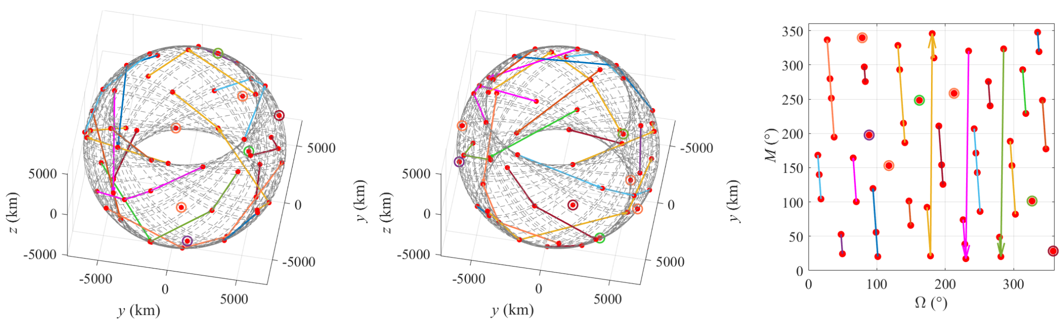
As shown in Fig. 4.10, T_m significantly alters the lower bound of f_1 (minimum launch



(a) $f_1 = 15,\ f_2 = 1390.43$ kg



(b) $f_1 = 20,\ f_2 = 867.56$ kg



(c) $f_1 = 25,\ f_2 = 692.56$ kg

Figure 4.9: Visualized Pareto-optimal L&D strategies for the 58-satellite case. Each subfigure depicts initial ECI orbital positions on the left and the $\Omega - M$ distribution on the right. Colored arrows connect satellites within the same launch, indicating their deployment order.

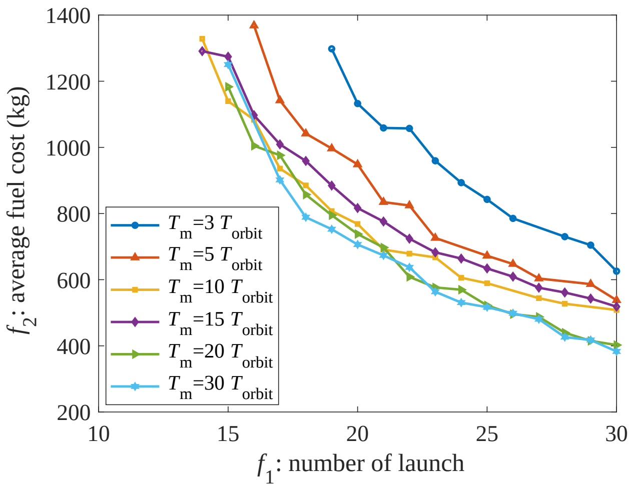


Figure 4.10: Pareto front comparison under different T_m of single-RGT regional coverage constellation with 58 satellites case.

count). Specifically, increasing T_m from 3 to 15 reduces the minimum required launches from 19 to 14. Moreover, for a fixed launch number, a longer T_m generally reduces the average fuel cost by up to 300 kg, albeit at the expense of total deployment time. Notably, the Pareto fronts stabilize when $T_m \geq 20$, indicating diminishing returns in fuel savings beyond this threshold. Therefore, setting T_m between 20 and 30 offers a practical balance between fuel efficiency and deployment speed.

4.5.2 Results on Large-Scale Walker LEO Constellation

To evaluate the proposed framework's performance on large-scale global coverage missions, a multi-shell Walker constellation scenario is designed, similar to the Starlink architecture [136]. This test case comprises 900 satellites distributed across three shells with varying altitudes (see Table 4.4). The spatial distribution is visualized in the $\Omega - M$ frame in Fig. 4.11. The maximum number of launches is constrained to 160, maintaining other simulation parameters consistent with Table 4.1.

Table 4.4: Configuration of the multi-shell Walker constellation test case.

	Orbit altitude/km	Inclination/°	Number of satellites	Number of planes	Phasing factor
Shell 1	600	60	360	30	12
Shell 2	900	60	300	30	12
Shell 3	1200	60	240	20	12

The proposed CNS-LSGA framework is benchmarked against population-based methods under identical launch constraints. Fig. 4.12 and Table 4.5 present the comparative results.

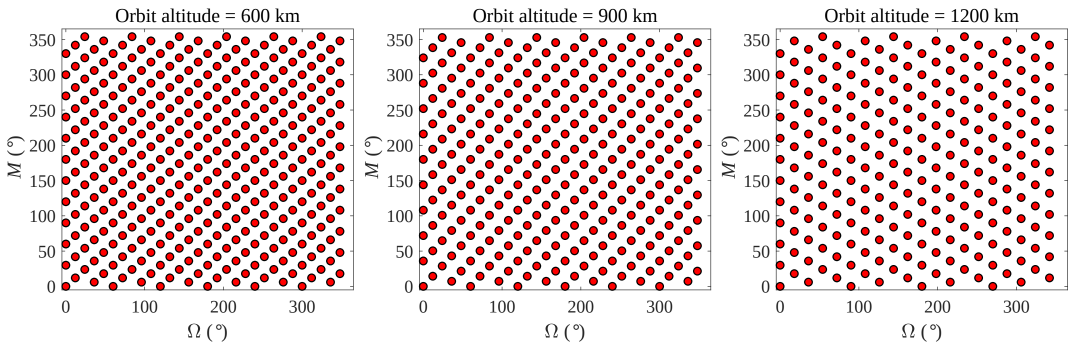


Figure 4.11: Satellite distribution of the multi-shell Walker constellation.

The simulation results confirm that the proposed ϵ-constraint CNS-LSGA method retains its performance advantage in large-scale heterogeneous scenarios. As indicated in Table 4.5, both MOEA/D and MSCMO failed to converge to feasible solutions. While NSGA-II produced 5 non-dominated solutions, our framework generated 38, demonstrating a substantial expansion of the feasible solution space. Specifically, the proposed method achieved a 23% reduction in minimal launch requirements (120 vs. 156) and lower fuel consumption compared to NSGA-II. Consequently, the HV metric improved by over 137% (0.1014 vs. 0.0427). Although the computational time increased to 1988.12 seconds—approximately double that of NSGA-II—due to the scale of the sub-problems,

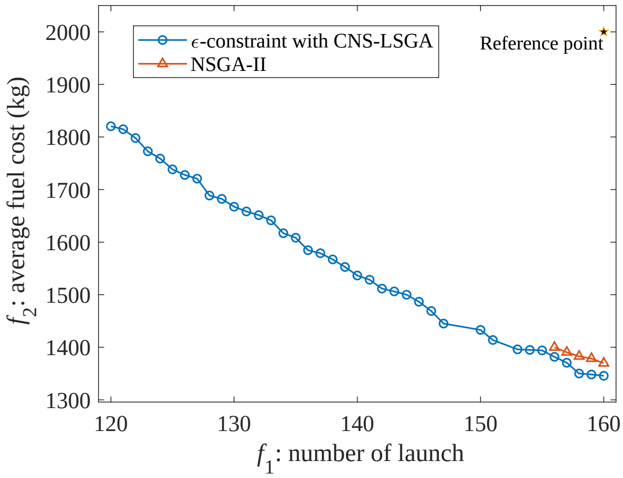


Figure 4.12: Pareto front comparison under the multi-shell Walker constellation test case.

Table 4.5: Quantitative comparison of L&D results for the multi-shell Walker constellation.

Constellation scale	Algorithm	CPU time	HV	Number of non-dominated solutions	$f_{1\,\min}$	$f_{1\,\max}$	$f_{2\,\min}$	$f_{2\,\max}$
900 satellites	ϵ-constraint with CNS-LSGA	1988.12 s	**0.1014**	**38**	**120**	160	**1345.87**	**1820.46**
	NSGA-II	941.71 s	0.0427	5	156	160	1369.71	1399.85
	MOEA/D	1007.17 s	/	/	/	/	/	/
	MSCMO	**838.18 s**	/	/	/	/	/	/

Reference point for calculating HV: [160,2000].

the significant gains in solution availability and optimality validate the enhanced robustness and scalability of the proposed approach.

4.5.3 Results on Regional Coverage Multi-RGT LEO Constellation

To further verify the framework's adaptability, this study examines a multi-RGT regional coverage constellation (see Table 3.2), comparing the resulting Pareto front against standard population-based multi-objective algorithms. This scenario involves 72 satellites spanning two distinct RGTs. Unlike single-RGT cases, the differing RAAN drift rates across RGTs cause the relative orbital geometry to vary over time. To address this time-dependency, a 10-day interval between consecutive launches is enforced, adding dynamic constraints to the L&D optimization. The orbital maneuver cost is determined using a combined maneuver method. Specifically, (1) the velocity increment for intra-RGT transfers is computed as:

$$\Delta V = \sqrt{(\Delta V_u/2)^2 + \Delta {V_\alpha}^2 + \Delta V_u \Delta V_\alpha \cos\frac{\pi+\alpha}{2}} + \frac{\Delta V_u}{2} \tag{4.21}$$

where ΔV_u is the velocity increment required for the in-plane phasing maneuver, ΔV_α is the velocity increment required for the plane change maneuver. α is the dihedral angle between the initial and target orbital planes. (2) maneuvers between satellites on different RGTs is calculated by:

$$\Delta V = \begin{cases} \sqrt{\|\Delta \boldsymbol{v}_{a1}\|^2 + \Delta {V_\alpha}^2 - 2\|\Delta \boldsymbol{v}_{a1}\|\Delta V_\alpha \cos\frac{\pi-\alpha}{2}} + \|\Delta \boldsymbol{v}_{a2}\|, & a_s > a_t \\ \|\Delta \boldsymbol{v}_{a1}\| + \sqrt{\|\Delta \boldsymbol{v}_{a2}\|^2 + \Delta {V_\alpha}^2 + 2\|\Delta \boldsymbol{v}_{a2}\|\Delta V_\alpha \cos\frac{\pi+\alpha}{2}}, & a_s < a_t \end{cases} \tag{4.22}$$

where $\|\Delta \boldsymbol{v}_{a1}\|$ and $\|\Delta \boldsymbol{v}_{a2}\|$ are the velocity increments required for the Hohmann maneuvers at the departure and arrival orbits, respectively. a_s and a_t are the semimajor axes of the initial and target orbits, respectively.

Furthermore, since the satellites are distributed across different RGTs, the differences in semimajor axis and inclination cannot be neglected. Accordingly, the coefficients in Eq. (4.20) are set as: $\lambda_1 = 0.5$, $\lambda_2 = 0.9$, $\lambda_3 = 0.9$, and $\lambda_4 = 0.05$.

The proposed framework is benchmarked against the selected population-based MOEAs, with outcomes detailed in Table 4.6 and Fig. 4.13 (maximum launches set to 40). Notably, the framework demonstrates high computational efficiency for medium-scale scenarios,

Table 4.6: L&D result comparison of the four algorithms under under multi-RGT regional coverage constellation test case

Constellation scale	Algorithm	CPU time	HV	Number of non-dominated solutions	$f_{1\,\min}$	$f_{1\,\max}$	$f_{2\,\min}$	$f_{2\,\max}$
72 satellites	ϵ-constraint with CNS-LSGA	**179.22 s**	**0.3748**	**24**	**16**	40	**540.44**	**1879.29**
	NSGA-II	181.04 s	0.2926	16	25	40	544.18	1089.03
	MOEA/D	281.47 s	0.0849	2	39	40	551.91	569.59
	MSCMO	201.37 s	0.3171	18	23	40	542.47	1161.49

Reference point for calculating HV: [40,2000].

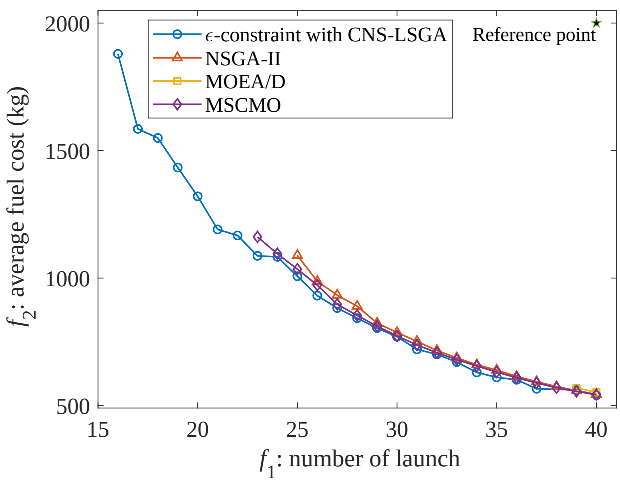


Figure 4.13: Pareto front comparison of the four algorithms under the multi-RGT regional coverage constellation test case.

completing execution in 179.22 seconds, which is competitive with or superior to the baseline algorithms.

As evidenced by the results, the ϵ-constraint CNS-LSGA achieves complete Pareto dominance over the comparison group. It significantly expands the feasible solution space, reducing the minimum launch requirement to 16 (versus 23 for the baselines) while maintaining superior fuel efficiency. Consequently, the proposed method yields 24 non-dominated solutions compared to a maximum of 18 by competitors, driving a 63% to 227% increase in the HV metric. These metrics confirm the algorithm's robustness and superiority in handling constrained L&D problems.

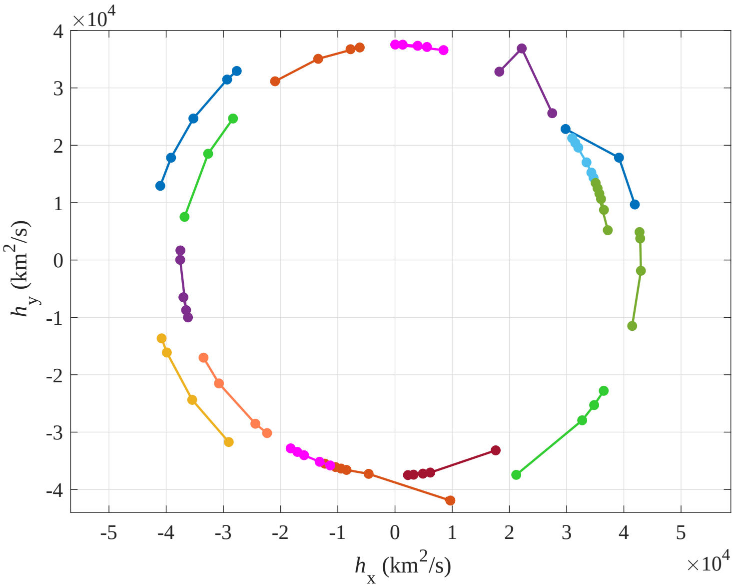


Figure 4.14: Visualizations of Pareto-optimal L&D results with 16 launches under the multi-RGT regional coverage constellation test case.

Fig. 4.14 depicts the optimal 16-launch strategy in the $h_x - h_y$ relative frame, employing the denser RGT 2 (inner circle) as a reference for RGT 1 satellites (outer circle). The visualization reveals that launch batches (distinguished by color) primarily cluster satellites within the same RGT based on RAAN proximity. This grouping strategy effectively minimizes high-consumption plane change maneuvers, thereby validating the rationale behind the distance metric coefficients selected for Eq. (4.20).

4.6 Summary

This chapter addresses the bi-objective combinatorial optimization of OTV-based LEO constellation L&D processes. By simultaneously optimizing satellite combinations and deployment sequences, the proposed model aims to minimize both total launches and average fuel consumption. To navigate the highly constrained solution space effectively, a specialized solution framework integrating the ϵ-constraint method with the CNS-LSGA is developed. This algorithm employs a Cluster and Nearest Sorting (CNS) initialization strategy alongside problem-specific local search operators to resolve feasibility issues and enhance solution quality. The efficacy of this approach is verified through extensive comparative experiments across various constellation configurations.

Simulation results yield two primary conclusions: (1) Ablation studies confirm that CNS initialization significantly bolsters algorithmic robustness across disparate scales, while local search operators improve performance under strict launch limitations. (2) The proposed hybrid framework exhibits superior stability and broader Pareto front coverage compared to conventional population-based multi-objective algorithms. Consequently, this study provides a robust quantitative tool for strategic L&D mission planning and cost benchmarking in constellation design.

Chapter 5

Mission Planning for LEO Constellation On-Orbit Servicing Systems

5.1 Introduction

The long-term operation of LEO constellations inevitably leads to satellite failures, which occupy valuable orbital resources and increase collision risks. On-orbit servicing (OOS) offers a solution by refueling fuel-depleted satellites to extend their lifespan or by de-orbiting non-operational ones to free up orbital slots. The complexity and scale of LEO constellations necessitate pre-determined mission scheduling to ensure OOS is efficient and cost-effective.

Traditional OOS modes are inadequate for the demands of LEO constellations. This chapter proposes a “service station-service satellite” architecture for long-duration OOS, addressing both refueling and disposal needs. The mission planning is decomposed into two coupled subproblems: service station orbit design and service satellite mission scheduling, which are co-optimized to maximize system efficiency.

The service station orbit design leverages the J_2 perturbation effect to manage the large RAAN differences among CSs. An optimal initial orbit allows the station to approach all assigned targets sequentially, reducing fuel consumption for the service satellites. For constellations requiring multiple service stations, this chapter proposes a two-stage ”clustering-optimization” method. First, an improved spectral clustering algorithm groups targets based on a modified orbital distance metric. Then, each cluster is assigned to a station, and the station's optimal orbit is determined via nonlinear programming.

Once station orbits are set, scheduling the service satellites is critical. This involves optimizing mission start times and assignments to avoid time conflicts and ensure adherence to fuel capacity constraints, as the fuel required for a mission is time-dependent. This chapter models the scheduling problem as a Mixed-Integer Nonlinear Programming (MINLP) problem, with time windows to narrow the search space. A Genetic Quantum Algorithm (GQA) is employed to efficiently find near-optimal solutions for this large-scale problem, optimizing both the number of service satellites and their total fuel consumption.

5.2 On-Orbit Servicing Mission Scenario Based on the "Service Station-Service Satellite" Mode

This section details the overall and single-mission procedures for OOS based on the "service station-service satellite" mode. It also outlines the objectives and constraints for mission planning within the OOS system. Furthermore, based on the single-mission procedure, this section introduces methods for estimating the orbital transfer cost for a service satellite performing a single OOS mission and for calculating this cost while considering mission execution time.

5.2.1 Overall On-Orbit Servicing Mission Procedure

As illustrated in Figure 5.1, some satellites in the LEO constellation are nearing fuel depletion (blue satellites), while others (red-shaded satellites) are non-operational due to payload or other critical component failures. These satellites are collectively referred to as Client Satellites (CS) in the OOS mission. The spacecraft that provides refueling or de-orbit device installation services is called a Service Station (SS). The SS is equipped with multiple identical service satellites, each capable of performing one OOS mission at a time. The service satellites are responsible for traveling between the SS and the CSs to deliver fuel or de-orbit devices. Each SS carries sufficient fuel and a limited number of service satellites. The primary goal of OOS is to manage these two types of satellites to minimize their impact on the overall performance of the constellation. This chapter defines two OOS mission for CSs based on their condition:

1) For fuel-depleted CSs, on-orbit refueling (OOR) is performed to restore their normal operation and extend their service life. Specifically, a service satellite travels from the SS to the CS, refuels it, and then returns to the SS.
2) For non-operational CSs, a controlled de-orbit process is initiated to free up valuable orbital resources for new satellite launches. Specifically, a service satellite docks with the non-operational CSs and installs a de-orbit propulsion device. After the service satellite undocks, the de-orbit device ignites, sending the non-operational CSs into a disposal orbit.

The set of all CSs is denoted as $C = \{c_1, c_2, ..., c_j, ..., c_N\}$, where N is the number of CSs, $j \in \{1, 2, ..., N\}$ is the index of the CS, and c_j is the j-th CS. φ_j represents the fuel demand or the mass of the de-orbit device for CS c_j. The initial orbital elements of c_j at the start of the mission are $E_{\mathrm{c}}^j = \{a_{\mathrm{c}}^j, e_{\mathrm{c}}^j, i_{\mathrm{c}}^j, \Omega_{\mathrm{c}}^j, \omega_{\mathrm{c}}^j, f_{\mathrm{c}}^j\}$. Multiple SSs are launched

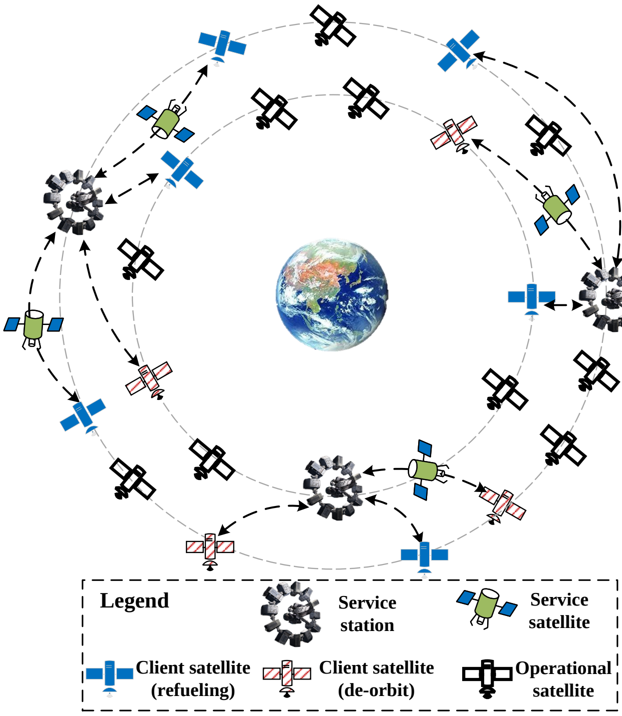


Figure 5.1: On-orbit servicing mission scenario based on the "service station-service satellite" mode [137].

into predetermined orbits. These orbits are designed to have a RAAN precession rate (drift rate) difference with the target constellation, allowing the service satellites to naturally approach each CS under the J_2 perturbation effect. The RAAN precession rate is calculated by Eq.(3.4). The set of SSs is denoted as $S = \{s_1, s_2, ..., s_i..., s_K\}$, where K is the number of SSs and s_i is the i-th SS. The initial orbital elements of s_i, $E_\mathrm{s}^i = \{a_\mathrm{s}^i, e_\mathrm{s}^i, i_\mathrm{s}^i, \Omega_\mathrm{s}^i, \omega_\mathrm{s}^i, f_\mathrm{s}^i\}$, are to be optimized. The service satellites are responsible for traveling between the SS and the CSs to deliver fuel and de-orbit devices. Each SS can carry a maximum of M service satellites, and v_i^k denotes the k-th service satellite of s_i. All service satellites are identical and have a fuel capacity of Φ. Due to this fuel capacity limit, each service satellite can only service one CS at a time. All OOS missions must be completed before the mission deadline T_f.

5.2.2 Single On-Orbit Servicing Mission Procedure

When the orbit of a SS precesses near the orbital plane of a CS, a service satellite departs from the station to rendezvous and dock with the target, returning after the OOS mission is complete. Since the relative orbital positions of the SS and the CS are constantly changing, the service satellite typically cannot depart and return at the exact moment their RAANs align. Instead, it must depart before this moment and return after. As shown in Figure 5.2, t_{ij} is the time when the RAAN difference between SS s_i and CS c_j becomes zero. The duration of each OOS mission is D. t_{e}^{ijk} is the departure time of the k-th service satellite from s_i to c_j, which is the mission start time. $t_{\mathrm{l}}^{ijk} = t_{\mathrm{e}}^{ijk} + D$ is the return time from c_j after the OOS mission is completed. To narrow the search space, a time window is defined for each OOS mission based on t_{ij}: $\mathrm{TW}_j = \left[tw_{\mathrm{e}}^{j}, tw_{\mathrm{l}}^{j}\right] = [t_{ij} - \delta, t_{ij} + \delta]$. Here, δ adjusts the length of the time window and must satisfy $\delta \geq D/2$. The execution period of each mission must fall within its corresponding time window.

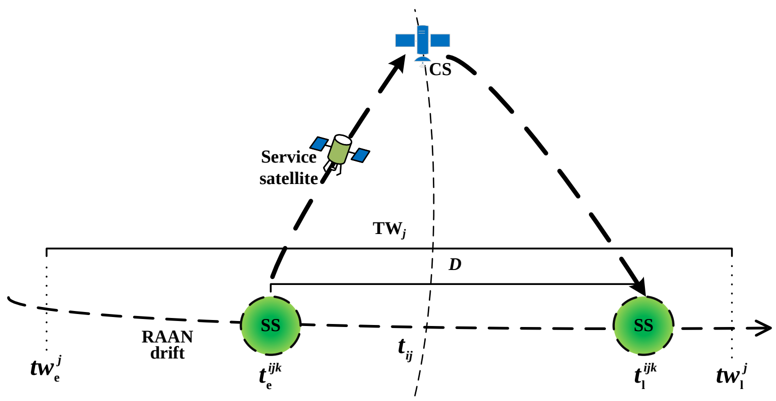


Figure 5.2: Single servicing mission procedure for one service satellite departing from SS and returning [137].

5.2.3 Objectives of Orbital Design and Mission Scheduling for On-Orbit Servicing Systems

The OOS mission planning problem considered in this chapter is based on the following assumptions:

Assumption 5.1. *Both CSs and SSs are in near-circular orbits with an eccentricity of 0.*

Assumption 5.2. *The remaining fuel of the CSs is sufficient for their routine orbit maintenance before the mission deadline.*

Assumption 5.3. *The fuel demand φ_j for each CS is known and fixed during the mission planning phase.*

The objective of OOS system mission planning is to minimize mission costs, including total fuel consumption and the number of service satellites used, while ensuring all OOS tasks are completed. To achieve this goal, the following aspects need to be optimized:

1) The assignment of CSs to each SS.
2) The initial orbital parameter design for each SS.
3) The assignment of service satellites for each OOS mission.
4) The start and end time scheduling for each OOS mission.

Additionally, several constraints must be satisfied:

1) The orbital altitude of each SS must meet safety requirements.
2) Each CS must be serviced exactly once.
3) Multiple OOS missions assigned to the same service satellite must not overlap in timeline.
4) All missions must be completed before the mission deadline T_{f}.

5) The fuel consumption for each mission must not exceed the service satellite's maximum fuel capacity $\varPhi$.
6) The number of service satellites per SS is limited to M.

5.2.4 Fuel Cost Calculation for a Single Round-Trip Orbit Maneuver of a Service Satellite

Suppose a service satellite v_i^k needs to maneuver from SS s_i (with orbital elements $E_\text{s}^i = \{a_\text{s}^i, i_\text{s}^i, \Omega_\text{s}^i\}$) to CS c_j (with orbital elements $E_\text{c}^j = \{a_\text{c}^j, i_\text{c}^j, \Omega_\text{c}^j\}$). The orbital transfer cost of the service satellite is the mass of fuel required for its orbital maneuvers. The initial mass of the service satellite can be expressed as

$$m = m_\text{dry} + \varphi_j + m_\text{fuel}^{ij} \tag{5.1}$$

where m_dry is the dry mass of the service satellite, and m_fuel^{ij} is the fuel mass consumed by the service satellite to maneuver from s_i to c_j.

The minimum fuel carried by the service satellite should be just enough to be depleted upon its return to the SS after completing the OOS mission. This leads to:

$$m_\text{dry} = m' \exp\left(\frac{-\Delta V_2^{ij}}{g_0 I_\text{sp}}\right) \tag{5.2}$$

where m' is the remaining mass of the service satellite after servicing the CS, g_0 is the standard gravitational acceleration, I_sp is the specific impulse of the service satellite's engine, and ΔV_2^{ij} is the velocity increment required for the service satellite to return to s_i.

For the first maneuver (from the SS to the CS), we have

$$m' + \varphi_j = m \exp\left(\frac{-\Delta V_1^{ij}}{g_0 I_{\text{sp}}}\right) \tag{5.3}$$

where ΔV_1^{ij} is the velocity increment required for the service satellite to maneuver from s_i to c_j. Combining Eqs. (5.1), (5.2), and (5.3), we get

$$m_{\text{fuel}}^{ij} = m_{\text{dry}}\left[\exp\left(\frac{\Delta V_{ij}}{g_0 I_{\text{sp}}}\right) - 1\right] + \varphi_j\left[\exp\left(\frac{\Delta V_1^{ij}}{g_0 I_{\text{sp}}}\right) - 1\right] \tag{5.4}$$

where $\Delta V_{ij} = \Delta V_1^{ij} + \Delta V_2^{ij}$.

In the SS orbit design problem, since the mission start times are not yet determined, the fuel consumption at the moment when the RAANs of the SS and the CS coincide is often used as an estimated maneuver cost. It is assumed that the service satellite departs and returns immediately at this time [90]. Therefore, we have $\Delta V_{ij} = 2\Delta V_1^{ij}$. The total estimated velocity increment for a service satellite departing from s_i to c_j and returning is

$$\Delta V_{ij} = 2(V_{\text{a}}^{ij} + \Delta V_{\alpha}^{ij}) \tag{5.5}$$

where V_{a}^{ij} and V_{α}^{ij} are the velocity increments required to eliminate the differences in semi-major axis and orbital plane between the SS and the CS, respectively. They can be calculated using standard Hohmann transfer and planar change without considering the RAAN difference [109].

In the service satellite mission scheduling problem, the velocity increment for an orbital maneuver depends on the start time and duration of the maneuver. Additionally, to improve calculation accuracy, the effect of J_2 perturbation must be considered. Therefore, this chapter uses the analytical rapid estimation method described in [138] to calculate the two maneuver velocity increments, ΔV_1^{ij} and ΔV_2^{ij}, which are related to t_e^{ijk} and t_l^{ijk}, respectively. The duration of each orbital maneuver is set to $t_d = 0.5$ days. Thus, the maneuver cost for a single OOS mission can be expressed as $F_{\mathrm{c}}\left(s_i, c_j, t_{\mathrm{e}}^{ijk}\right) = m_{\mathrm{fuel}}^{ij}\left(a_{\mathrm{s}}^i, a_{\mathrm{c}}^j, I_{\mathrm{s}}^i, I_{\mathrm{c}}^j, \Omega_{\mathrm{s}}^i, \Omega_{\mathrm{c}}^j, \varphi_j, t_{\mathrm{e}}^{ijk}\right)$.

5.3 Optimal Deployment Orbit Design of Service Stations

This section introduces the Modified Spectral Clustering-Nonlinear Programming (MSC-NLP) method for designing the orbits of multiple SSs. In the SS orbit design problem, the orbital elements a, i, and Ω must be determined, as they represent the shape and spatial orientation of the SS orbit. Hereafter, these three orbital elements will be used to describe the orbits of both SSs and CSs.

Previous research [89] has shown that deploying an SS in a region with a dense distribution of CSs can significantly reduce its orbital maneuver costs. Therefore, to solve the multi-SS orbit design problem, the CSs are first clustered based on their orbital distribution. Each cluster of CSs is then assigned to a single SS for on-orbit servicing (OOS). Subsequently, the orbital parameters of the corresponding SS are optimized based on the orbital distribution of the CSs within each cluster. Figure 5.3 illustrates this process.

The similarity between CS orbits cannot be effectively measured by common clustering distance metrics, such as Euclidean distance. Therefore, a modified spectral clustering method is employed to allow for flexible definition of an arbitrary distance metric, thereby enabling the clustering of CSs. The optimal orbit design problem is then modeled as a nonlinear programming (NLP) problem to minimize the estimated maneuver cost between the SS and the CSs. The model also incorporates multiple constraints, including SS orbital altitude safety, RAAN precession, and mission deadlines.

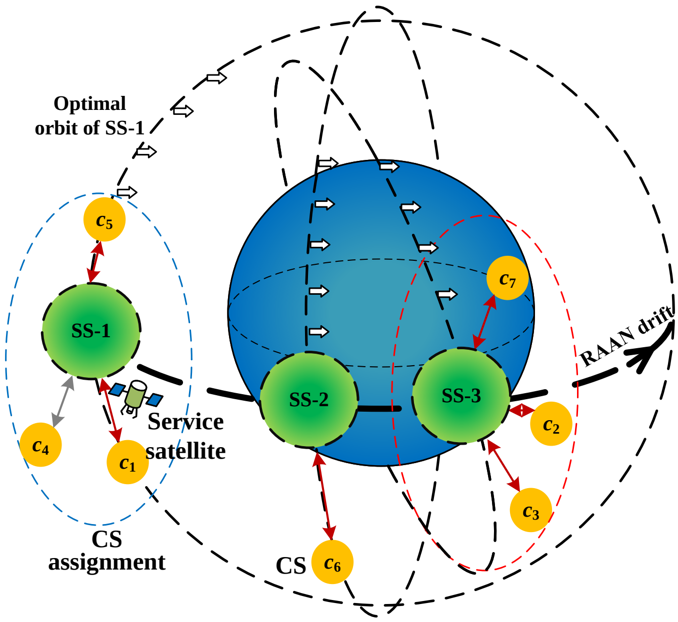
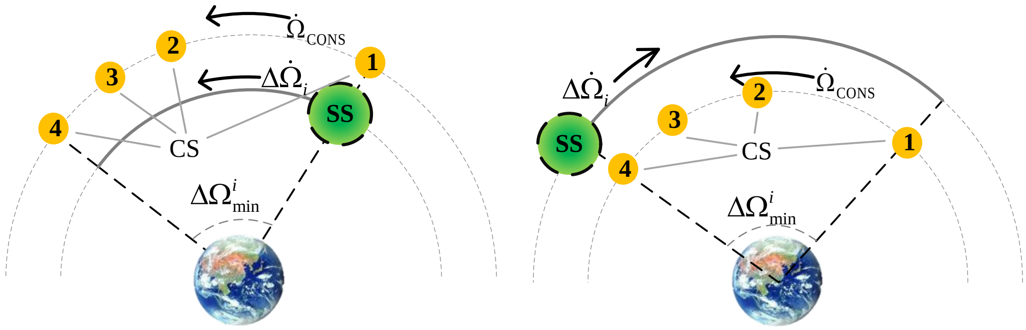


Figure 5.3: Optimal orbit design of SS based on CS assignment result.

5.3.1 Relative Drift Strategy of Service Stations and Client Constellation

Suppose an SS, s_i, is assigned to service a group of CSs, $C_i = \{c_1, c_2, ..., c_m, ..., c_{N_i}\}$, with N_i being the number of CSs. The corresponding set of RAANs is denoted as $R_i = \{\Omega_1, \Omega_2, ..., \Omega_m, ..., \Omega_{N_i}\}$, where the elements of C_i and R_i are in one-to-one correspondence. The elements in R_i satisfy: (1) $\Omega_m \in [0, 360°)$ for $m = 1, 2, ..., N_i$; and (2) for any $m > n$, $\Omega_m, \Omega_n \in R_i$, we have $\Omega_m \geq \Omega_n$. As shown in Figure 5.4, to ensure that the SS can sequentially approach all target CSs through RAAN precession, it is necessary to properly design the initial RAAN Ω_{s}^i, precession rate $\dot{\Omega}_{\mathrm{s}}^i$, and relative precession angle $\Delta\Omega_i$ (i.e., the RAAN precession of the SS relative to the CS constellation's orbit during the OOS mission period). To complete the mission as quickly as possible and minimize the difference between the SS orbit and the CS constellation's orbit, the minimum required $\Delta\Omega_i$ should be determined based on the RAAN distribution of the selected CSs.



(a) $\Delta\dot{\Omega}_i > 0$ (b) $\Delta\dot{\Omega}_i < 0$

Figure 5.4: Two cases of relative RAAN drift between SS and CS

For any two RAANs, $\Omega_m, \Omega_n \in R_i$, the RAAN difference in the counter-clockwise direction is defined as [137]:

$$\Delta\Omega_{\text{ac}}^{mn} = \begin{cases} 360^\circ - |\Omega_m - \Omega_n|, & m, n \in \{1, N_i\}, m \neq n \\ |\Omega_m - \Omega_n|, & \text{otherwise} \end{cases} \tag{5.6}$$

By mapping all angles in R_i onto a unit circle, the central angle corresponding to the minimum arc that contains all points is defined as the minimum enveloping RAAN angle for the CS group C_i, denoted as $\Delta\Omega^i_{\text{min}}$. It can be calculated as follows:

$$\Delta\Omega^i_{\text{min}} = 360^\circ - \max\left\{\Delta\Omega_{\text{ac}}^{mn}, \Delta\Omega_{\text{ac}}^{N_i 1}\right\}, \ m = 1, 2, ..., N_i - 1, \ n = m + 1 \tag{5.7}$$

Therefore, the minimum $\Delta\Omega_i$ is $\Delta\Omega^i_{\text{min}}$. The shortest time required for the SS to sequentially approach all assigned CSs is:

$$t_i = \frac{\Delta\Omega^i_{\text{min}}}{|\Delta\dot{\Omega}_i|} \tag{5.8}$$

where $\Delta\dot{\Omega}_i = \dot{\Omega}^i_{\text{s}} - \dot{\Omega}_{\text{CONS}}$, and $\dot{\Omega}^i_{\text{s}}$ is the RAAN precession rate of SS s_i. The initial RAAN of the SS can be chosen as one of the two elements forming the minimum enveloping arc. Let $\Delta\Omega^{kl}_{\text{ac}} = \max\left\{\Delta\Omega^{mn}_{\text{ac}}, \Delta\Omega^{N_i 1}_{\text{ac}}\right\}$, where $m = 1, 2, ..., (N_i - 1)$, $n = m + 1$, and k, l are the corresponding element indices. As shown in Figure 5.4, if the designed $\Delta\dot{\Omega}_i > 0$, the initial RAAN is $\Omega^i_{\text{s}} = \Omega_l$; otherwise, $\Omega^i_{\text{s}} = \Omega_k$.

5.3.2 Client Satellite Assignment by Modified Spectral Clustering

As illustrated in Figure 5.5, the basic idea of spectral clustering is to represent all clustering objects as an undirected graph and transform the clustering problem into an optimal graph partitioning problem [139]. Let the undirected graph corresponding to the CS set C be $G = (V, E)$, where $V = \{v_1, v_2, ..., v_N\}$, and $v_i \in V$ represents the i-th CS, c_i. The edge weight w_{ij} between v_i and v_j measures the similarity between the two vertices, with $w_{ij} \geq 0$. Considering that K SSs will be deployed, the cut weight for partitioning the graph into K subgraphs is defined as:

$$\text{cut}(A_1, A_2, ..., A_K) = \frac{1}{2}\sum_{i=1}^{K} \text{cut}(A_i, \bar{A}_i) \tag{5.9}$$

where $\text{cut}(A_i, \bar{A}_i) = \sum_{i \in A_i, j \in \bar{A}_i} w_{ij}$, A_i is the i-th subgraph, and $\bar{A}_i$ is its complement. The factor of $1/2$ ensures that each edge is counted only once. By minimizing the objective function (5.9), the optimal clustering result can be obtained. To further balance the number of vertices in each subset, Hagen et al. [140] proposed the following new objective function:

$$\text{RadioCut}(A_1, A_2, ..., A_K) = \sum_{i=1}^{K} \frac{\text{cut}(A_i, \bar{A}_i)}{|A_i|} \tag{5.10}$$

where $|A_i|$ is the number of vertices in subgraph A_i.

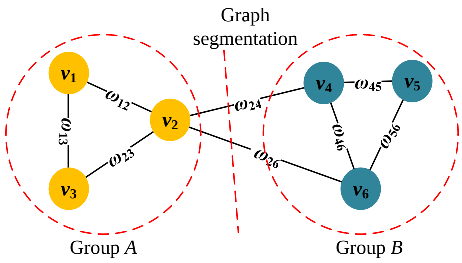


Figure 5.5: Diagram of Graph Segmentation in Spectral Clustering.

To solve the above problem, two distance metrics between any two vertices v_i, v_j are defined based on the orbital elements of the CSs:

$$r_{\Delta V} = \Delta V_{\text{a}}^{ij} + \Delta V_{\alpha}^{ij} \tag{5.11}$$

$$r_{\Omega} = \begin{cases} |\Omega_i - \Omega_j|, & |\Omega_i - \Omega_j| \leq 180^\circ \\ 360^\circ - |\Omega_i - \Omega_j|, & |\Omega_i - \Omega_j| > 180^\circ \end{cases} \tag{5.12}$$

As shown in Figure 5.6, $r_{\Delta V}$ represents the velocity increment for an orbital maneuver between two CSs, without considering the RAAN difference. This metric helps to reduce the differences in semi-major axis and orbital inclination within the clustered CS groups, thereby indirectly lowering the fuel consumption in the SS orbit design. r_{Ω} is the RAAN difference between two CSs. Incorporating r_{Ω} into the clustering distance metric helps to reduce the minimum enveloping RAAN angle $\Delta\Omega_{\text{min}}$ of the clustered CS group, which in turn reduces the relative precession rate between the SS orbit and the CS constellation, indirectly decreasing fuel consumption. These two distance metrics can be combined into a single comprehensive index [137]:

$$r_{ij} = (1-\lambda)r'_{\Delta V} + \lambda r'_{\Omega} \tag{5.13}$$

where $r'_{\Delta V}$ and r'_{Ω} denote the normalized values of $r_{\Delta V}$ and r_{Ω}, respectively. The weighting parameter λ balances the contribution of RAAN differences and maneuvering costs within the metric. Since the choice of λ influences the clustering outcome and the subsequent SS fuel efficiency, it depends heavily on the spatial distribution of the CSs; generally, a smaller λ is preferable when the targets share similar orbital altitudes.

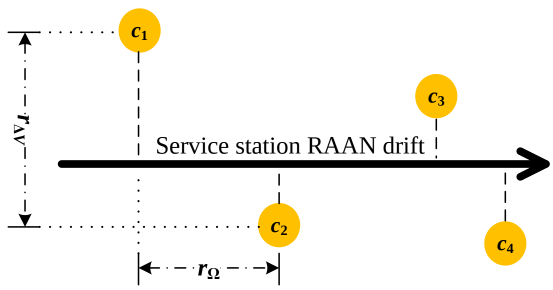


Figure 5.6: Schematic diagram of cluster distance metric between any two client satellites.

Next, to ensure that every vertex is connected and the graph structure is not overly dense, the k-Nearest Neighbors (kNN) method [141] is used to construct the similarity between any two vertices:

$$w_{ij} = w_{ji} = \begin{cases} 0, & v_i \notin k\,\mathrm{NN}(v_j) \text{ and } v_j \notin k\,\mathrm{NN}(v_i) \\ \exp\left(-\frac{r_{ij}^2}{(2\sigma)^2}\right), & v_i \in k\,\mathrm{NN}(v_j) \text{ or } v_j \in k\,\mathrm{NN}(v_i) \end{cases} \tag{5.14}$$

where $k\,\mathrm{NN}(v_i)$ represents the set of k nearest neighbors of v_i determined by (5.13). The edge weight between two connected nodes is weighted using a Gaussian-like kernel function. σ is a specified kernel scale factor. A smaller value of σ makes the similarity more sensitive to the distance between vertices, and vice versa. In this chapter, σ is set to 0.5. This allows for the definition of the similarity matrix $\boldsymbol{W} = [w_{ij}]^{N\times N}$. Let d_i be the degree of v_i:

$$d_i = \sum_{j=1}^{N} w_{ij}. \tag{5.15}$$

The degree matrix $\boldsymbol{D}$ can be expressed as $\boldsymbol{D} = \text{diag}(d_1, d_2, ..., d_N)$. Finally, the Laplacian matrix is defined as:

$$\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W} \tag{5.16}$$

Define the indicator vector for subgraph A_i as $\boldsymbol{h}_i = [h_{1i}, h_{2i}, ..., h_{ni}, ..., h_{Ni}]^{\mathrm{T}}$, where

$$h_{ni} = \begin{cases} \frac{1}{\sqrt{|A_i|}}, & v_n \in A_i \\ 0, & v_n \notin A_i \end{cases} \tag{5.17}$$

Based on the above equation, the indicator matrix can be further defined as $\boldsymbol{H} = [\boldsymbol{h}_1, \boldsymbol{h}_2, ..., \boldsymbol{h}_i, ..., \boldsymbol{h}_K]$. Since each vertex can only be assigned to one subgraph, we have $\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H} = \boldsymbol{I}$.

According to Eqs. (5.9) and (5.17), we can derive:

$$\begin{aligned}
\boldsymbol{h}_i^{\mathrm{T}}\boldsymbol{L}\boldsymbol{h}_i &= \frac{1}{2}\sum_{a=1}^{N}\sum_{b=1}^{N} w_{ab}(h_{ai} - h_{bi})^2 \\
&= \frac{1}{2}\left[\sum_{a\in A_i, b\notin A_i} w_{ab}\left(\frac{1}{\sqrt{|A_i|}} - 0\right)^2 + \sum_{a\notin A_i, b\in A_i} w_{ab}\left(0 - \frac{1}{\sqrt{|A_i|}}\right)^2\right] \\
&= \frac{1}{2}\left[\frac{\text{cut}(A_i, \bar{A}_i)}{|A_i|} + \frac{\text{cut}(\bar{A}_i, A_i)}{|A_i|}\right] = \frac{\text{cut}(A_i, \bar{A}_i)}{|A_i|}
\end{aligned} \tag{5.18}$$

Therefore, Eq. (5.10) can be rewritten as:

$$\text{RatioCut}(A_1, A_2, \cdots, A_K) = \sum_{i=1}^{K} \boldsymbol{h}_i^{\mathrm{T}} \boldsymbol{L} \boldsymbol{h}_i = \text{Tr}(\boldsymbol{H}^{\mathrm{T}} \boldsymbol{L} \boldsymbol{H}) \tag{5.19}$$

The problem is further transformed into:

$$\begin{cases} \min_{\boldsymbol{H} \in \mathbb{R}^{N \times K}} \text{Tr}(\boldsymbol{H}^{\mathrm{T}} \boldsymbol{L} \boldsymbol{H}) \\ \quad \text{s.t.} \quad \boldsymbol{H}^{\mathrm{T}} \boldsymbol{H} = \boldsymbol{I} \end{cases} \tag{5.20}$$

According to the Rayleigh-Ritz theorem [142], the optimal solution to this problem is a combination of the K smallest eigenvectors of $\boldsymbol{L}$. Let $\boldsymbol{U}$ be the matrix composed of the K smallest eigenvectors of $\boldsymbol{L}$. By applying the k-means algorithm to each row of $\boldsymbol{U}$, it is partitioned into K groups, where the i-th row of $\boldsymbol{U}$ corresponds to vertex v_i. This yields the optimal assignment of the CS set C, denoted as $\{C_1, C_2, ..., C_i, ..., C_K\}$. The number of CSs in C_i is denoted as N_i.

5.3.3 Multiple Service Station Orbital Design by Nonlinear Programming

After completing the CS assignment, the optimal orbital parameters for the SS corresponding to each CS cluster need to be determined.

Let a subset of CSs obtained through clustering be $C_i = \{c_1, c_2, ..., c_j, ..., c_{N_i}\}$, where the orbital elements for each CS c_j are $\{a_{\mathrm{c}}^j, i_{\mathrm{c}}^j, \Omega_{\mathrm{c}}^j\}$. The SS s_i will be deployed to provide OOS for the CSs in C_i. The optimal SS orbit design problem can be formulated as follows:

(1) Decision Variables

1) a_{s}^i: Semi-major axis of the SS;

2) i_{s}^{i}: Orbital inclination of the SS;

where $i = 1, ..., K$. Additionally, the initial RAAN of the SS, Ω_{s}^{i}, can be determined using the method described in Section 5.3.1.

(2) Optimization Objective

$$\min \sum_{i=1}^{K} \sum_{j=1}^{N_i} m_{\text{fuel}}^{ij} \left(a_{\text{s}}^{i}, a_{\text{c}}^{j}, i_{\text{s}}^{i}, i_{\text{c}}^{j}, \varphi_j\right) \tag{5.21}$$

The above equation aims to minimize the total estimated fuel consumption for all SSs, where the fuel consumption only accounts for the orbital transfer of the service satellites. In the equation, K is the number of SSs, and N_i is the number of CSs assigned to SS i.

(3) Constraints

$$\frac{\Delta\Omega_{\text{min}}^{i}}{|\Delta\dot{\Omega}_i|} \leq T_f, \quad \forall i = 1, ..., K \tag{5.22}$$

$$a_{\text{min}} \leq a_{\text{s}}^{i} \leq a_{\text{max}}, \quad \forall i = 1, ..., K \tag{5.23}$$

$$m_{\text{fuel}}^{ij} \left(a_{\text{s}}^{i}, a_{\text{c}}^{j}, i_{\text{s}}^{i}, i_{\text{c}}^{j}, \varphi_j\right) + \hat{\varphi}_j \leq \varPhi, \quad \forall i = 1, ..., K, \ j = 1, ..., N_i \tag{5.24}$$

Constraint (5.22) requires that the SS must pass by all assigned CSs through precession before the mission deadline T_f, where $\Delta\Omega_{\text{min}}^{i}$ is the minimum enveloping RAAN angle of the CS group, and $\Delta\dot{\Omega}_i$ is the RAAN precession rate difference between the SS and the CS constellation's orbit. Constraint (5.23) limits the orbital altitude of the SS to balance launch vehicle capacity and orbital safety. Constraint (5.24) ensures that the fuel used for a single mission by a service satellite does not exceed its maximum fuel capacity. Here, $\hat{\varphi}_j$ represents the payload delivered to target c_j. If the service satellite performs an on-orbit refueling (OOR) mission, then $\hat{\varphi}_j = \varphi_j$; if it performs a de-orbit device installation mission, then $\hat{\varphi}_j = 0$. This problem is a nonlinear programming problem with inequality constraints, which can be solved using the *fmincon* function in MATLAB. The initial values of the decision variables are set to the average of the orbital elements of the CS set corresponding to the SS.

5.4 Mission Scheduling of Service Satellites

This section proposes a Mixed-Integer Nonlinear Programming (MINLP) model for service satellite mission scheduling, which simultaneously optimizes fuel consumption and the number of service satellites. To simplify the handling of different types of decision variables and to ensure high algorithmic performance, this chapter employs a Genetic Quantum Algorithm (GQA) to solve the problem.

5.4.1 Modeling the Mission Scheduling Problem for Service Satellites

Due to the extended duration of OOS operations, it is impractical for a service satellite to execute both transfer legs exactly when the RAAN difference between the SS and CS aligns to zero. Consequently, maneuver timings must be optimized relative to the SS precession rate to minimize propellant usage. Moreover, to avoid temporal conflicts and ensure efficient resource allocation among multiple targets, the specific assignment of service satellites must be predetermined. The MINLP formulation for scheduling the assets of a single SS s_i is presented as follows:

(1) Decision Variables

1) x_{ijk}: A binary variable indicating the assignment of a service satellite to a CS mis-

sion. It is defined as:

$$x_{ijk} = \begin{cases} 1, & \text{if CS } c_j \text{ is serviced by service satellite } v_i^k \text{ from SS } s_i \\ 0, & \text{otherwise} \end{cases} \tag{5.25}$$

2) t_{e}^{ijk}: The mission start time. It represents the start time for service satellite v_i^k from SS s_i to perform the OOS mission for CS c_j.

In the above decision variables, $i = 1, 2, ..., K$, $j = 1, 2, ..., N_i$, and $k = 1, 2, ..., M$.

(2) Optimization Objectives

$$f_1\left(x_{ijk}, t_{\mathrm{e}}^{ijk}\right) = \min \sum_{j=1}^{N_i} \sum_{k=1}^{M} x_{ijk} m_{\text{fuel}}^{ij}\left(a_{\mathrm{s}}^i, a_{\mathrm{c}}^j, i_{\mathrm{s}}^i, i_{\mathrm{c}}^j, \Omega_{\mathrm{s}}^i, \Omega_{\mathrm{c}}^j, \varphi_j, t_{\mathrm{e}}^{ijk}\right) \tag{5.26}$$

$$f_2\left(x_{ijk}, t_{\mathrm{e}}^{ijk}\right) = \min \sum_{k=1}^{M} \frac{\sum_{j=1}^{N_i} x_{ijk}}{\max\left(\sum_{j=1}^{N_i} x_{ijk}, 1\right)} \tag{5.27}$$

Objective function (5.26) aims to minimize the total fuel consumption of all service satellites within SS s_i, where m_{fuel}^{ij} includes the fuel for both departure from and return to the SS. Objective function (5.27) is to minimize the number of service satellites required.

(3) Constraints

$$\sum_{k=1}^{M} x_{ijk} = 1,\ \forall j = 1, 2, ..., N_i \tag{5.28}$$

$$tw_{\mathrm{e}}^{j} \le t_{\mathrm{e}}^{ijk} < t_{\mathrm{l}}^{ijk} \le tw_{\mathrm{l}}^{j} \tag{5.29}$$

$$\begin{cases} t_{\mathrm{e}}^{imk} > t_{\mathrm{l}}^{ink},\ \text{if}\ t_{\mathrm{e}}^{imk} > t_{\mathrm{e}}^{ink} \\ t_{\mathrm{l}}^{imk} < t_{\mathrm{e}}^{ink},\ \text{if}\ t_{\mathrm{e}}^{imk} < t_{\mathrm{e}}^{ink} \end{cases} \tag{5.30}$$

$$\forall m, n = 1, ..., N_i,\ m \neq n,\ x_{ink} = x_{imk} = 1$$

$$m_{\text{fuel}}^{ij} + \hat{\varphi}_j \le \Phi,\ \forall x_{ijk} = 1 \tag{5.31}$$

Constraint (5.28) ensures that each CS is serviced by exactly one service satellite. Constraint (5.29) requires that each OOS mission must be completed within its designated time window. Constraint (5.30) prevents time overlaps between multiple OOS missions assigned to the same service satellite. Constraint (5.31) is the fuel capacity constraint for the service satellite, which has the same meaning as in Eq. (5.24), but the fuel calculation method is different.

5.4.2 Solving the Mission Scheduling Problem with a Genetic Quantum Algorithm

The Genetic Quantum Algorithm (GQA) was first proposed by Han and Kim [143]. In GQA, the genes of each chromosome are represented by Q-bits, and the population is updated using a quantum gate mechanism inspired by quantum computing. These features give GQA advantages such as a large search space, small population size, and fast convergence. On one hand, GQA can find near-optimal solutions even with a small population size [143]. On the other hand, GQA operates directly on Q-bits, and its population update mechanism is independent of the encoding scheme, which simplifies the algorithm design process.

(1) Decision Variable Encoding Mechanism

This chapter encodes the two types of decision variables, x_{ijk} and t_{e}^{ijk}, separately. A discrete real-number encoding is used for x_{ijk}, with a chromosome length of N_i. For the j-th gene g_j^x, we have $1 \leq g_j^x \leq M, g_j^x \in \mathbb{Z}$. g_j^x represents the index of the service satellite assigned to CS c_j. That is, if $g_j^x = g$, then $x_{ijg} = 1$, and $x_{ijk} = 0$ for all other $k \neq g$, satisfying constraint (5.28). A continuous real-number encoding is used for t_{e}^{ijk}, also with a chromosome length of N_i. To satisfy constraint (5.31), each g_j^t must satisfy $0 \leq g_j^t \leq tw_{\mathrm{l}}^j - tw_{\mathrm{e}}^j - D$. g_j^t represents the start time of the mission for CS c_j relative to the beginning of its time window, regardless of which service satellite performs it.

In GQA, each gene is represented by a Q-bit. A Q-bit is defined as a pair of numbers $[\alpha, \beta]^{\mathrm{T}}$, where $\alpha^2 + \beta^2 = 1$. Therefore, the j-th Q-bit can be represented as a point on the unit circle: $[\alpha_j, \beta_j]^{\mathrm{T}} = [\cos\psi_j, \sin\psi_j]^{\mathrm{T}}$, where ψ_j is the angle between the point, the origin, and the x-axis.

The original chromosome genes g_j^x and g_j^t must be mapped from the Q-bits. The conversion from a Q-bit to g_j^x is:

$$g_j^x = \lceil M\alpha_j^2 \rceil \tag{5.32}$$

The conversion from a Q-bit to g_j^t is:

$$g_j^t = tw_{\mathrm{e}}^j - \alpha_j^2 (tw_{\mathrm{l}}^j - tw_{\mathrm{e}}^j) \tag{5.33}$$

(2) Population Update Mechanism Based on Quantum Gates

The population is updated by updating the Q-bits using quantum gates. Subsequently, the entire population is updated based on the mapping relationship between Q-bits and chromosome genes. This chapter uses two types of quantum gates.

First, in each iteration, a rotation gate is used to update the Q-bits. For the j-th Q-bit in a quantum chromosome, the update is performed as follows:

$$\begin{bmatrix} \alpha_j' \\ \beta_j' \end{bmatrix} = \begin{bmatrix} \cos\Delta\psi_j & -\sin\Delta\psi_j \\ \sin\Delta\psi_j & \cos\Delta\psi_j \end{bmatrix} \begin{bmatrix} \alpha_j \\ \beta_j \end{bmatrix} \tag{5.34}$$

where $\left[\alpha_j', \beta_j'\right]^{\mathrm{T}}$ is the updated Q-bit, $\Delta\psi_j$ is the rotation angle of the gate. According to [144], $\Delta\psi_j$ can be calculated as:

$$\begin{cases} \Delta\psi_j = \Delta\psi_0 C(l) e^{\frac{l}{L}} \\ C(l) = \eta\left[1 - C(l-1)\right] C(l-1) \\ C(l) \neq 0.5, C(l) \in [0,1] \end{cases} \tag{5.35}$$

where l is the rank of the current chromosome sorted by fitness, L is the population size, and η is a chaotic coefficient. In this chapter, we set $\eta = 4$ and $\Delta\psi_0 = 0.0785$. The direction of rotation is determined by:

$$Q = \begin{vmatrix} \alpha_i^o & \alpha_i \\ \beta_i^o & \beta_i \end{vmatrix}$$

where $[\alpha_i^o, \beta_i^o]^{\mathrm{T}}$ is the i-th Q-bit of the current best individual. When $Q \neq 0$, the rotation direction is $-\mathrm{sgn}(Q)$; otherwise, the direction is arbitrary.

To prevent getting stuck in local optima, a NOT gate is also introduced for individual updates. For each chromosome in the population, some Q-bits are randomly selected to undergo a NOT gate operation, which is expressed as:

$$\begin{bmatrix} \alpha'_j \\ \beta'_j \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \alpha_j \\ \beta_j \end{bmatrix} \tag{5.36}$$

(3) Fitness Calculation and Constraint Penalties

A penalty function method is used to handle constraints. Constraints (5.28) and (5.29) are satisfied by the encoding mechanism, while constraints (5.30) and (5.31) need to be checked. Therefore, the fitness function is defined as:

$$F_{\text{fit}} = \sum_{j=1}^{N_i} \sum_{k=1}^{M} k x_{ijk} m_{\text{fuel}}^{ij} + Pen_1 p_1 + Pen_2 p_2 \tag{5.37}$$

where k is the service satellite index, p_1 and p_2 are the penalty functions for constraints (5.30) and (5.31) respectively, and Pen_1 and Pen_2 are positive penalty coefficients. The factor k acts as a penalty for the number of service satellites used, thereby optimizing objective f_2. The penalty terms p_1 and p_2 are specifically defined as:

$$\begin{cases} p_1 = \sum\limits_{k=1}^{M} \sum\limits_{m,n \in C_i, m \neq n} x_{imk} x_{ink} t_{\text{over}}^{mn} / 2 \\ p_2 = \sum\limits_{k=1}^{M} \sum\limits_{j=1}^{N_i} x_{ijk} \max \left\{ m_{\text{fuel}}^{ij} + \varphi_j - \Phi, 0 \right\} \end{cases} \tag{5.38}$$

where t_{over}^{mn} is the duration of the time overlap between the missions for CSs c_m and c_n.

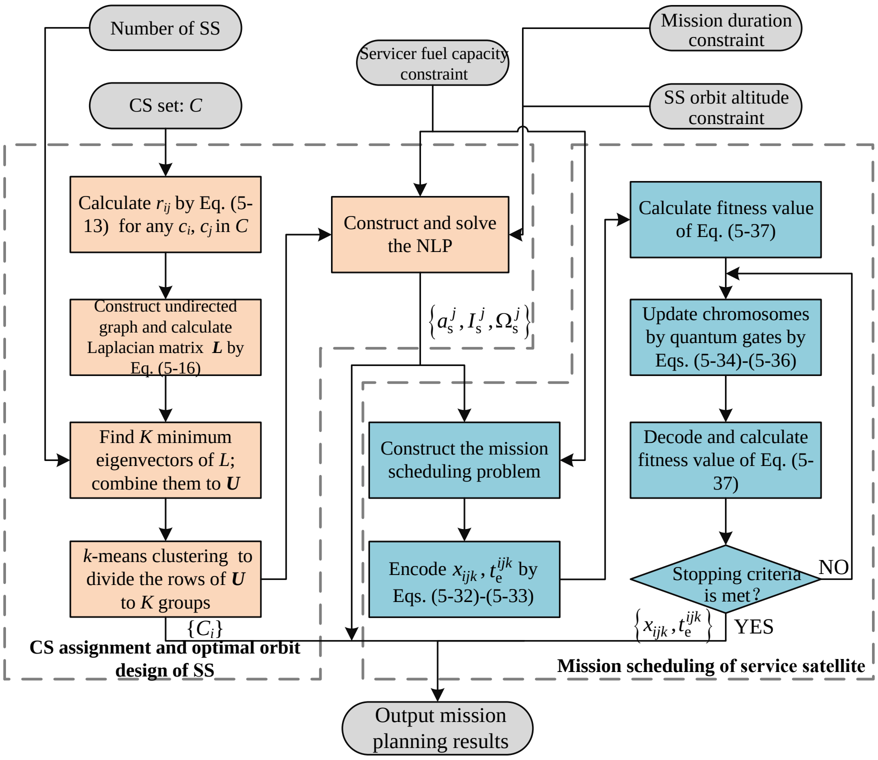


Figure 5.7: Overall optimization framework of orbital design and mission scheduling for OOS systems [137].

5.5 Overall Procedure of Orbital Design and Mission Scheduling for OOS Systems.

The overall procedure for OOS system mission planning is divided into two stages: SS orbit design and service satellite mission scheduling. The first stage involves the optimal orbit design of the SSs based on the CS clustering results. The second stage performs service satellite mission scheduling based on the obtained optimal orbit parameters. The two stages are interdependent, as the results of the first stage directly influence the mission scheduling in the second stage. This two-stage optimization framework is illustrated in Figure 5.7. The inputs to this framework include the set of CSs, the number of SSs, and other constraint parameters. The outputs include the clustering results of the CSs, the corresponding SS orbital parameters, the start time for each OOS mission, and the index of the responsible service satellite. Specifically, the first stage first clusters the CSs based on a relative distance metric and then designs the optimal orbital parameters for the SSs according to the clustering results. The second stage then arranges the mission schedule for the service satellites based on the optimal orbital parameters obtained in the first stage, determining the start time and responsible service satellite for each OOS mission. The two stages are closely related, as the orbit design results from the first stage affect the subsequent mission scheduling arrangements.

5.6 Simulation of Case Study

The results of SS orbit design and mission scheduling for an OOS system are constrained by the distribution of CSs. To fully validate the effectiveness and superiority of the proposed orbit design and mission scheduling methods, this section first conducts a feasibility verification based on the multi-RGT orbit regional coverage constellation designed in Chapter 3. Subsequently, two case studies of Sun-Synchronous Orbit (SSO) Earth observation constellations with different scales and complex orbital distributions are investigated[1]. CSs in SSO are typically distributed at different altitudes and inclinations, with large RAAN differences, which leads to significant variations in fuel consumption to reach different targets. Therefore, the SSO constellation distribution provides a challenging scenario for validating the proposed method. The RAAN precession rate of SSO is $\dot{\Omega}_{\text{SSO}} = 0.9856°/\text{day}$. First, a comparative experiment is conducted on a small-scale case from existing literature to verify the effectiveness and superiority of the method in

[1] Test constellation parameter can be found in: https://github.com/HITPengHan/OOR_mission_planning

this chapter. Then, a simulation based on a larger set of CSs is performed to validate the scalability of the proposed method. Furthermore, the performance of the GQA algorithm is compared with other optimization algorithms in a large-scale scenario. Other parameters related to the SSs, service satellites, and mission scenarios for the three test cases are shown in Table 5.1. All simulations were performed on a laptop with an Intel Core i7-12700H CPU and 16 GB of RAM.

Table 5.1: Parameter settings of service satellite and mission scenario for the three test scenarios.

Parameter	Regional Coverage Constellation	Small-Scale SSO Constellation	Large-Scale SSO Constellation
Number of CSs	72	20	100
Number of controlled de-orbit CSs	7	0	14
Mass of de-orbit device	180 kg	/	170 kg
Number of SSs K	5	3	6
Max number of service satellites per SS M		5	
SS semi-major axis limit		$R_e + 300 \text{ km} \leq a_s^i \leq R_e + 3000 \text{ km}$	
Service satellite dry mass m_{dry}		150 kg	
Service satellite thruster specific impulse I_{sp}		326.53 s	
Service satellite fuel capacity Φ		500 kg	
Single OOS mission duration D		30 days	
Overall OOS mission period T_f	360 days	760 days	760 days

$R_e = 6378.137$ km is the Earth's radius.

5.6.1 Feasibility Verification in Mission Scenario of Multi-Regional Coverage Constellation

This section verifies the feasibility of the proposed mission planning framework using the multi-regressive orbit regional coverage constellation designed in Table 3.2 of Chapter 3 as the OOS target. All 72 satellites in this constellation are set as CSs. To reflect a realistic scenario, the vast majority are designated for on-orbit refueling (OOR), while 7 CSs are selected for controlled de-orbit operations. The fuel demand for OOR targets is randomly set between 180 kg and 230 kg, and the mass of the de-orbit device for controlled de-orbit targets is set to 180 kg. There are 5 SSs, each capable of carrying 5 service satellites. The remaining parameters are shown in Table 5.1.

5.6.1.1 Result of CS Assignment and SS Optimal Orbit Design

Since different regressive orbits have different precession rates, clustering satellites from different regressive orbits into the same group could result in the subsequently designed SS being unable to pass by all CSs with different precession rates within the specified time. Therefore, the clustering process should set the value of λ to be small (finally set to 0.1) to encourage clustering results that only include CSs from the same regressive orbit. The CS clustering results are shown in Figure 5.8. The results show that the 72 CSs are divided into 5 clusters, corresponding to 5 SSs. The CSs within each cluster are all on the same regressive orbit: $C_1 - C_3$ are satellites on regressive orbit 2, and $C_4 - C_5$ are satellites on regressive orbit 1. According to the precession trajectories of the SSs, each SS can pass by its corresponding CSs within the specified mission period, providing opportunities for OOS. The optimal orbit design results for the SSs are presented in Table 5.2. This table details the orbital elements for each SS, including its precession rate, semi-major axis, inclination, and initial RAAN, alongside the corresponding CS cluster and estimated fuel consumption. The RAAN precession rate of each SS differs slightly from that of the target CSs, but all five SSs precess in the same direction. Each SS maintains an orbital altitude below 3000 km, adhering to the specified constraints. These results confirm that the proposed orbit design method is effective for designing SS orbits for a multi-regressive orbit regional coverage constellation.

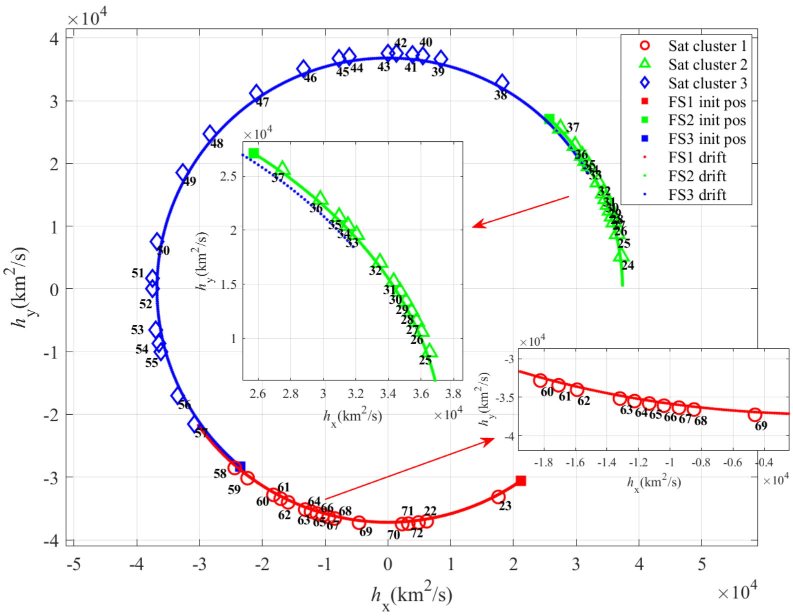
Table 5.2: Results of CS clustering and optimal orbit design of corresponding SSs (72 CSs and 5 SSs scenario).

SS	$\dot{\Omega}_s^i$ (°/day)	$\dot{\Omega}_{\text{CONS}}$ (°/day)	a_s^i (km)	i_s^i (°)	Ω_s^i (°)	$\Delta\Omega_i$ (°)	CS Cluster	Number of CSs	Estimated Fuel Consumption (kg)
s_1	-3.59	-3.365	7852.77	41.68	34.80	68.50	{23,22,72,71, 70,69,68,67, 66,65,64,63, 62,61,60,59, 58}	17	180.00
s_2	-3.48	-3.365	7923.83	41.68	136.46	35.00	{37,36,35,34, 33,32,31,30, 29,28,27,26, 25,24}	14	77.29
s_3	-3.88	-3.365	7683.40	41.68	320.35	154.00	{57,56,55,54, 53,52,51,50, 49,43,48,47, 46,45,44,42, 41,40,39,38}	20	469.04
s_4	-3.28	-2.915	7737.47	49.61	125.55	110.50	{9,8,7,6,5,4, 3,2,1,21,20}	11	215.72
s_5	-3.40	-2.915	7659.59	49.61	305.10	146.00	{19,18,17,16, 15,14,13,12, 11,10}	10	257.35
Total	/	/	/	/	/	/	/	72	1199.40

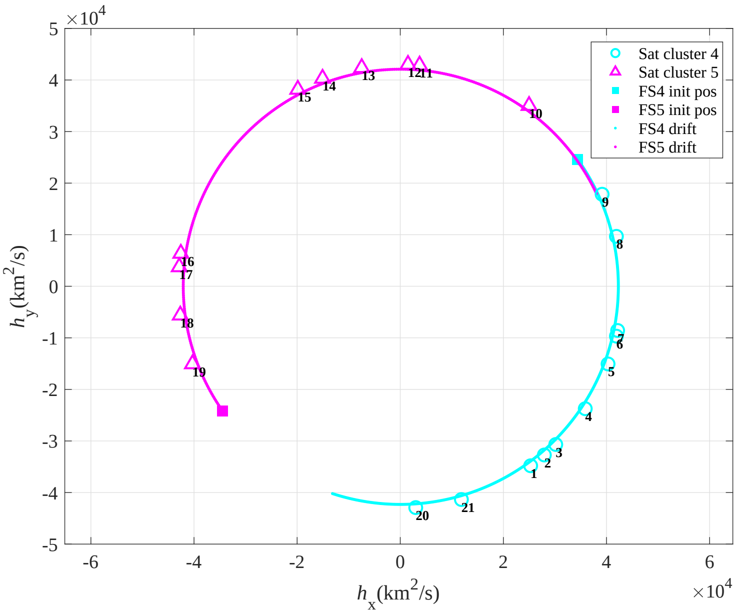
5.6.1.2 Result of Service Satellite Mission Scheduling

The mission scheduling process, which constitutes the second stage of the mission planning framework, is primarily solved by the GQA. In the GQA, the population size is set to 40, and the maximum number of iterations is 5000, which also serves as the termination condition. The two penalty coefficients Pen_1 and Pen_2 in Eq. (5.37) are set to 500 (kg/day) and 100, respectively.

The results of the service satellite mission scheduling are shown in Table 5.3. The table lists the total fuel consumption, the number of service satellites used, the fitness value, and the time constraint violation for each SS. All SSs are able to complete their OOS missions for the target CSs within the mission period. The total fuel consumption is 3666.41 kg, and a total of 13 service satellites are used. The mission scheduling fuel consumption is



(a) Distribution of cluster C_1-C_3 and the corresponding SSs



(b) Distribution of cluster C_4-C_5 and the corresponding SSs

Figure 5.8: Distribution of CS clusters and orbit drift of SSs (72 CSs and 5 SSs scenario).

Table 5.3: Service satellite mission scheduling results by GQA (72 CSs and 5 SSs scenario).

SS	Total Fuel Consumption (kg)	Number of Service Satellites Used	Fitness Value	Time Constraint Violation
s_1	496.4934	4	905.2527	[0,0,0,0]
s_2	199.9681	2	271.3791	[0,0]
s_3	1493.5556	3	2532.8642	[0,0,0]
s_4	643.8118	2	760.8097	[0,0]
s_5	832.5785	2	996.3893	[0,0]
Total	3666.4073	13	5466.6951	0

approximately three times the estimated fuel value in Table 5.2. This indicates that there are significant differences between the service satellite orbits and the target constellation orbits, and the dense mission schedule forces service satellites to perform orbital maneuvers before the SS precesses to the target, thus increasing fuel consumption. However, the overall fuel consumption remains within a reasonable range, as also shown in Figure 5.10. The number of service satellites used by each SS does not exceed its carrying capacity, further demonstrating the reasonableness of the mission scheduling results.

Figure 5.9 displays the OOS mission timeline for each service satellite in each SS. In the figure, each mission is marked with a colored block, the length of which includes both the OOS mission execution time and the orbital transfer time. Colorless blocks represent controlled de-orbit missions. Colored blocks represent OOR missions, with different colors corresponding to different SSs. The vertical axis indicates the service satellite number for each timeline. As can be seen, there are no time conflicts between any two consecutive missions for a single service satellite, and all missions are completed before the mission deadline, satisfying the time constraints.

Figure 5.10 shows the fuel consumption for each OOS mission as determined by the mission scheduling. For each mission, the light-colored portion represents the fuel for

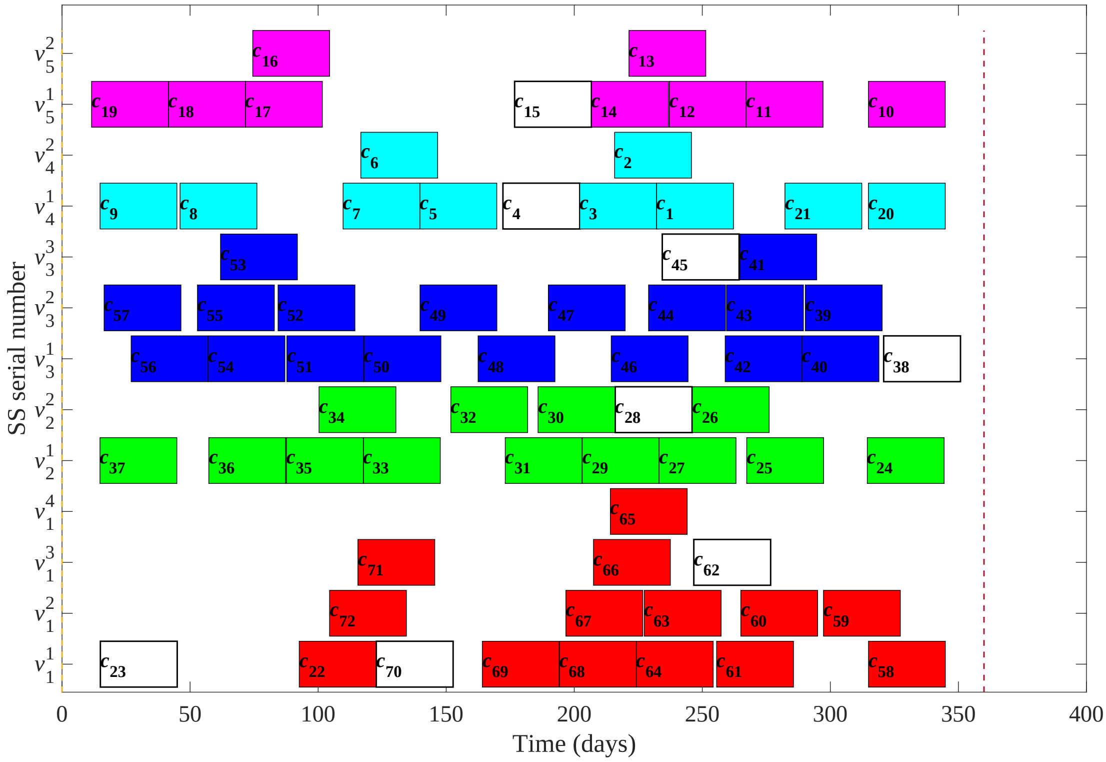


Figure 5.9: OOS mission operation timeline of each service satellite in each SS (72 CSs and 5 SSs scenario).

orbital transfer, while the dark-colored portion represents the fuel delivered to the target. If a mission does not have a dark-colored portion, it is a controlled de-orbit mission. Different colors correspond to different SSs. The fuel consumption for all missions does not exceed the maximum fuel capacity of the service satellites. The fuel consumption varies significantly among different SSs, which is closely related to the distribution of CSs and the orbit design of the SSs. Within the same SS, the fuel consumption for orbital transfers shows little variation. This is because the CSs assigned to a single SS are on the same regressive orbit, sharing the same semi-major axis and inclination. When the SS naturally precesses near a CS, the main orbital differences are in semi-major axis and inclination. Therefore, the fuel required to transfer from the SS to the CS is nearly identical, with mi-

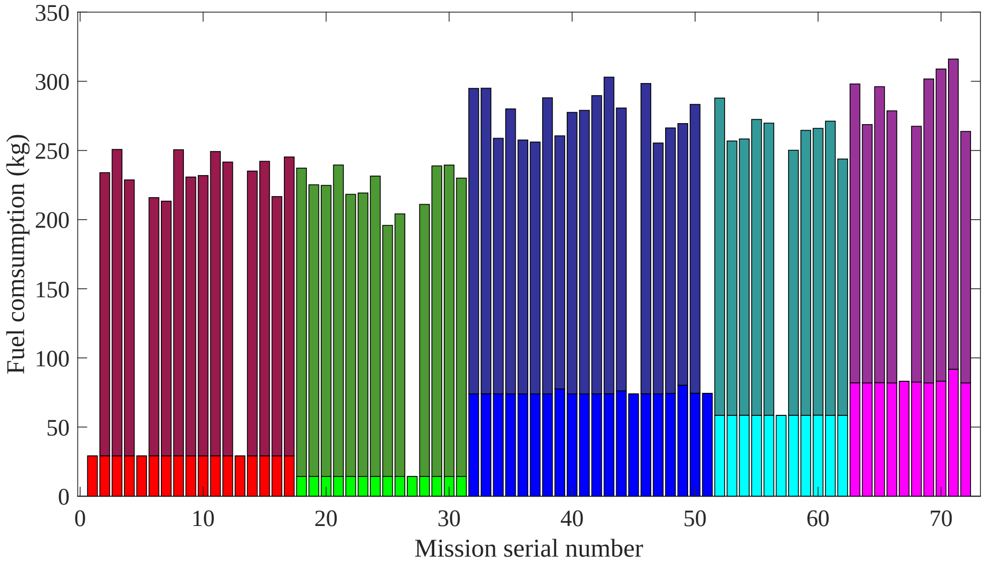


Figure 5.10: Fuel cost for each OOS mission obtained by service satellite mission scheduling (72 CSs and 5 SSs scenario).

nor differences arising from the varying initial mass of the service satellite due to different fuel demands of the CSs.

5.6.2 Comparative Study Based on Small-Scale SSO Constellation Mission Scenario

This section selects the SSO satellite OOR mission scenario from the literature [90] as a comparative case study and compares the results with those presented in that paper. The scenario includes 20 CSs distributed in SSO, with orbital altitudes uniformly distributed between 300 and 1700 km.

5.6.2.1 Result of CS Assignment and SS Optimal Orbit Design

By tuning the value of λ in Eq. (5.13) from 0 to 1, the optimal value was determined to be 0.1, as shown in Figure 5.11. The subsequent CS assignment and SS orbit design results are presented in Table 5.4, Figure 5.12, and Figure 5.13.

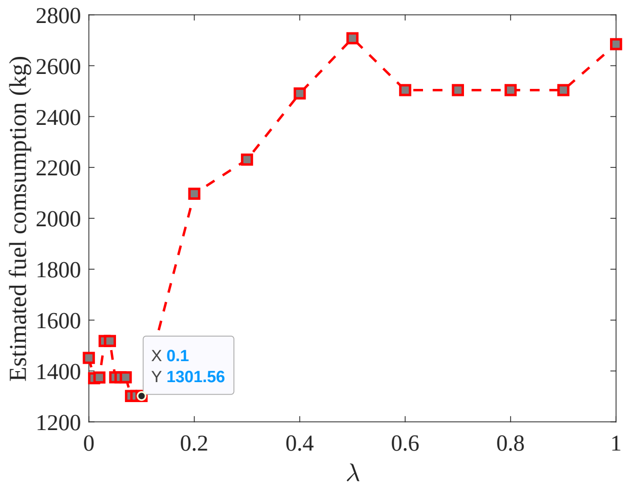


Figure 5.11: Optimal value tuning of λ (20 CSs and 3 SSs scenario).

Table 5.4: Results of CS clustering and optimal orbit design of corresponding SSs (20 CSs and 3 SSs scenario).

SS	$\dot{\Omega}_s^i$ (°/day)	a_s^i (km)	i_s^i (°)	Ω_s^i (°)	$\Delta\Omega_i$ (°)	CS Cluster	Number of CSs	Estimated Fuel Consumption (kg)
s_1	1.17	7200.10	100.35	143.42	130.10	{7,11,8,10, 9,12}	6	252.56
s_2	1.33	6678.14	99.02	183.79	240.60	{6,1,4,2,3,5}	6	340.46
s_3	1.33	7985.01	107.10	320.43	244.20	{16,13,15,17, 19,18,14,20}	8	708.54
Total	/	/	/	/	/	/	20	1301.56

Table 5.4 presents the results of CS assignment and optimal SS orbit design. The total estimated fuel consumption is 1301.56 kg, which is a 16.5% reduction compared to the 1558 kg reported in [90]. This improvement is attributed to two main factors. First, the

concept of the minimum enveloping RAAN angle, $\Delta\Omega_{\min}$, for CS clusters was introduced, which reduces the RAAN precession path of the SS for a given group of CSs. Second, an improved spectral clustering method was used to assign all CSs to multiple SSs. The new clustering distance metric considers both the velocity increment and the RAAN difference between CSs. These two improvements effectively reduce the relative precession rate and orbital differences between the CSs and the SSO, thereby lowering the estimated fuel consumption for transfers from the SS to each CS.

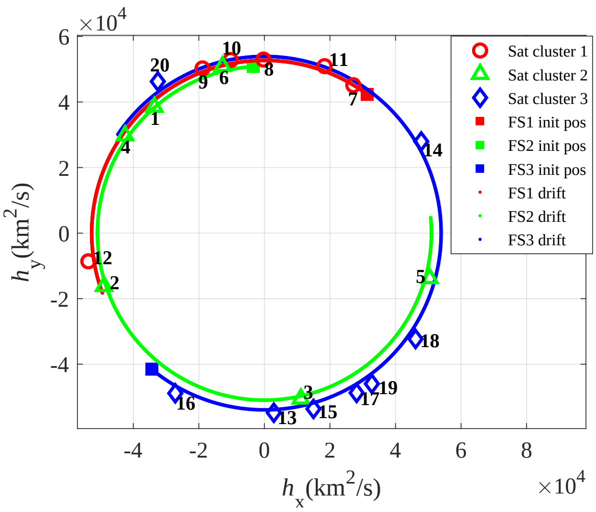


Figure 5.12: Distribution of CS clusters and orbit drift of SSs (20 CSs and 3 SSs scenario).

Figure 5.12 shows the projection of the orbital angular momentum of all SSs and CSs on the $h_x - h_y$ plane during the mission period. This coordinate system is stationary relative to the SSO, so the trajectories of the SSs reflect their precession relative to the SSO. The figure shows that each SS can approach all assigned CSs within its cluster during the mission period. Figure 5.13 further details the estimated fuel consumption for each OOR mission. Each bar in the figure corresponds to an SS of the same color in Figure 5.12. The light-colored portion of the bar represents the fuel required for orbital maneuvers, while the dark-colored portion represents the fuel demand of the CS. As shown in Figure 5.13, all OOR missions satisfy the fuel capacity constraint of the service satellites.

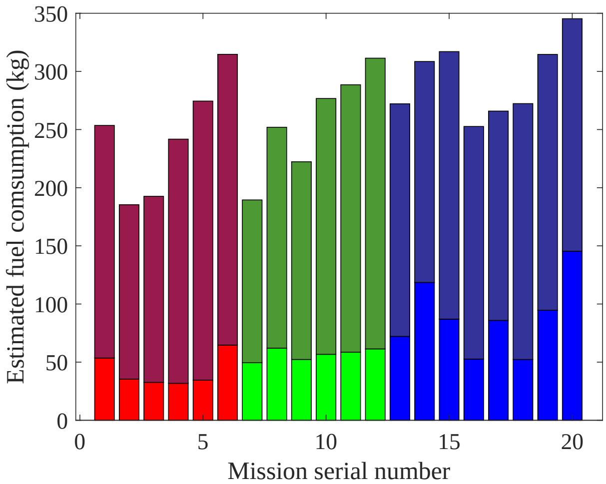


Figure 5.13: Estimated fuel costs of each OOR mission (20 CSs and 3 SSs scenario).

To further validate the performance of the proposed framework under different mission durations and compare it with the results from [90], the mission duration was varied from 760 days to 1860 days in steps of 100 days. The results are shown in Figure 5.14. As the mission duration increases, the total estimated fuel consumption gradually decreases. For the same mission duration, the MSC-NLP algorithm consistently outperforms the algorithm in [90].

The results also indicate that fuel consumption is closely related to the relative RAAN precession rate, $\Delta\dot{\Omega}$, between the SS and the CSs. A longer mission duration allows the optimized SS orbit to achieve a lower $\Delta\dot{\Omega}$, which reduces the difference between the SS orbit and the SSO constellation, thereby lowering the fuel consumption for orbital maneuvers. Similarly, for a fixed mission duration, fuel consumption and the SS's $\Delta\dot{\Omega}$ can be further reduced by appropriately selecting the RAAN span of the CSs. This further demonstrates that introducing the concept of the minimum enveloping angle in the CS assignment process can indirectly improve the effectiveness of the SS orbit design.

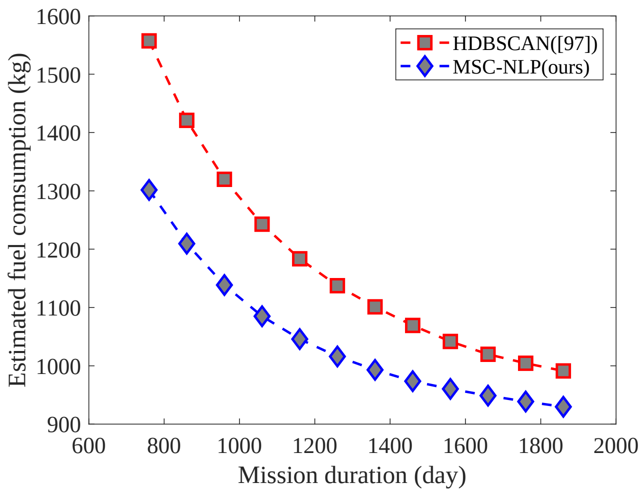


Figure 5.14: Comparison of estimated fuel costs in different mission duration (20 CSs and 3 SSs scenario).

5.6.2.2 Result of Service Satellite Mission Scheduling

In this section, the GQA parameters are the same as in Section 5.6.1.2. The mission scheduling results are shown in Table 5.5 and Figure 5.15.

Table 5.5: Comparison of service satellite mission scheduling results (20 CSs and 3 SSs scenario).

SS	Total Fuel Consumption (kg)		Number of Service Satellites Used	
	GQA	GCM[90]	GQA	GCM[90]
s_1	**261.83**	308.98	1	1
s_2	**514.21**	606.19	1	1
s_3	**699.90**	834.90	**1**	2
Total	**1475.96**	1750.07	**3**	4

In [90], a Gantt Chart Method (GCM) is used for mission scheduling, where the mission start time is set to the moment the RAAN difference between the SS and the CS is zero. If any two missions overlap in time, GCM assigns the subsequent mission to a new service satellite. Table 5.5 compares the mission scheduling results of the two algorithms.

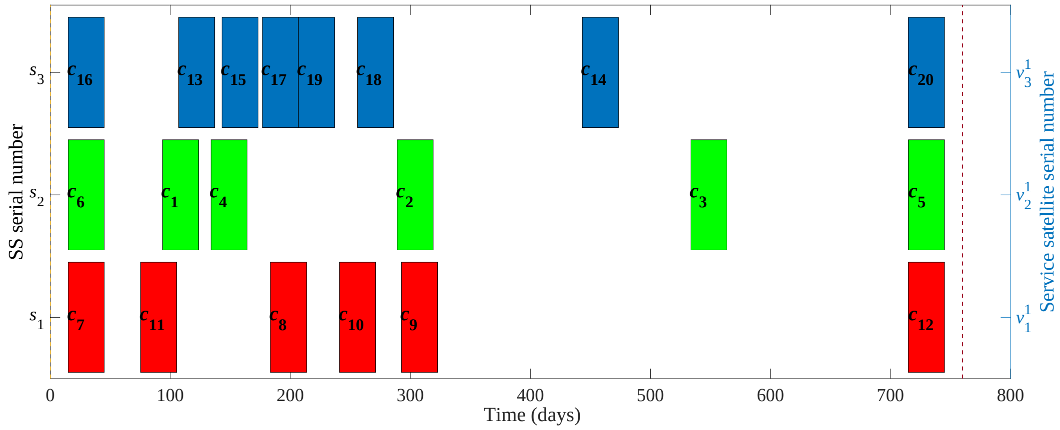


Figure 5.15: OOR mission operation timeline of each service satellite in each SS (20 CSs and 3 SSs scenario).

The GCM method results in a total fuel consumption of 1750 kg, while the GQA method reduces it to 1475.96 kg, a 15% decrease. Furthermore, GQA requires only 3 service satellites, fewer than the GCM result. These comparisons highlight the necessity of mission scheduling optimization and the effectiveness of the GQA method.

Figure 5.15 shows the mission execution timeline obtained by GQA. The results indicate that each SS needs only one service satellite to complete all OOR missions, with no time overlaps between any two missions, and all CSs are refueled before the mission deadline. Figure 5.16 shows the fuel consumption for each OOR mission. Compared to Figure 5.13, the orbital maneuver costs here are calculated more realistically, considering the start time of the maneuver and the effect of J_2 perturbation. All missions satisfy the service satellite's fuel capacity constraint.

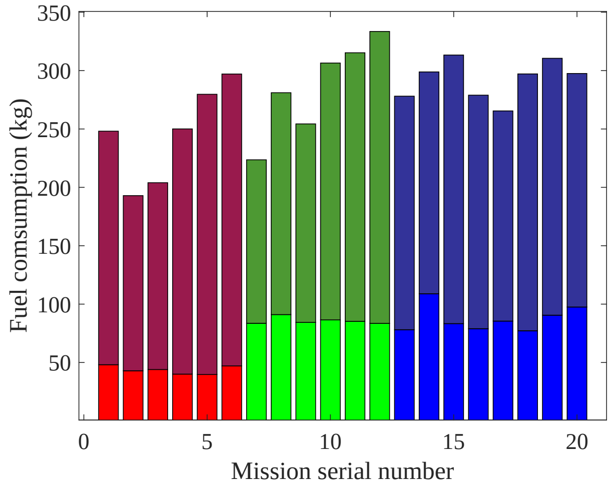


Figure 5.16: Fuel cost for each OOR mission obtained by service satellite mission scheduling (20 CSs and 3 SSs scenario).

5.6.3 Validation of Effectiveness under Large-Scale SSO Constellation Mission Scenario

This section validates the framework's scalability using a large-scale scenario with 100 CSs and 6 SSs. The CS orbital parameters are randomly generated, with some RAANs clustered near 0° and 90°, as shown in Figure 5.6.3. Among those CSs, 14 CSs are randomly designated for de-orbit missions (170 kg device), while the rest require refueling (fuel deamnd randomly set between 170-230 kg). Parameters are consistent with the small-scale case, but the scaling factor λ is fine-tuned to an optimal value of 0.06 for this larger problem, as shown in Figure 5.18.

Figure 5.19 and Table 5.6 show the CS assignment and optimal SS orbit design results. As shown in Figure 5.19, all CSs are assigned to an SS for OOS. The optimized SS orbits ensure that each SS can approach its assigned CSs before the mission deadline. Additionally, the relative precession rate of s_2 is $\Delta\dot{\Omega}_s^2 < 0$, which is also verified in Table 5.6. The

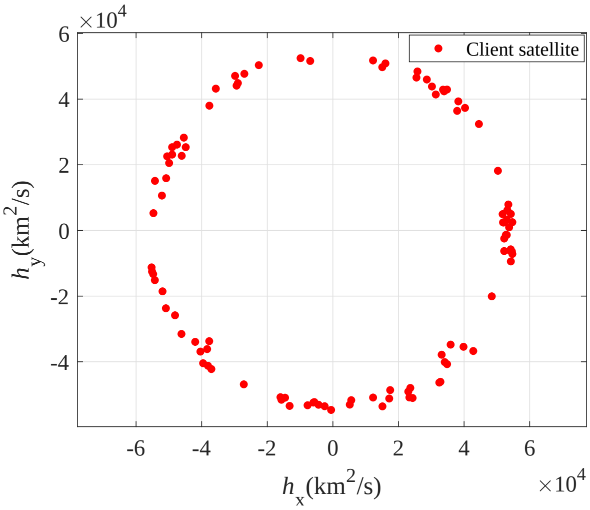


Figure 5.17: Satellite distribution of the SSO constellation in the $h_x - h_y$ frame.

total estimated fuel consumption is 4681.94 kg, which is a reasonable result.

To further demonstrate the performance of GQA in mission scheduling, several swarm intelligence optimization algorithms commonly used for OOS mission planning were selected for comparison, including GA [83], ACO [89], and Particle PSO [77]. All algorithms were configured with a population size of 40 and 5000 iterations (same as GQA), with other parameters set to the defaults in PlatEMO. Each algorithm was run 20 times independently, and the statistical results are shown in Table 5.7. The detailed results corresponding to the best fitness value obtained by each algorithm are presented in Table 5.8.

Table 5.7 shows that GQA significantly outperforms the other three intelligent optimization algorithms in terms of fitness evaluation metrics (average, minimum, and standard deviation), demonstrating superior search capability and result stability. This is mainly due to GQA's quantum-bit encoding, which can represent both continuous variables (mission start time) and discrete variables (service satellite assignment), and its quantum rotation gate operation, which effectively explores the solution space even with a

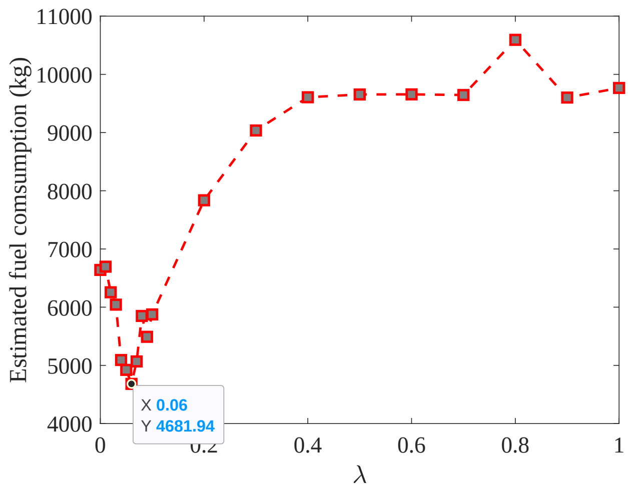
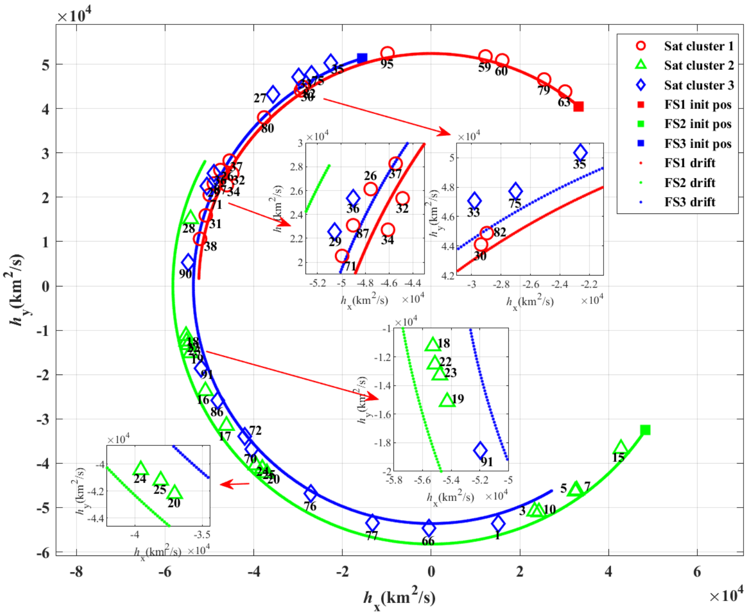


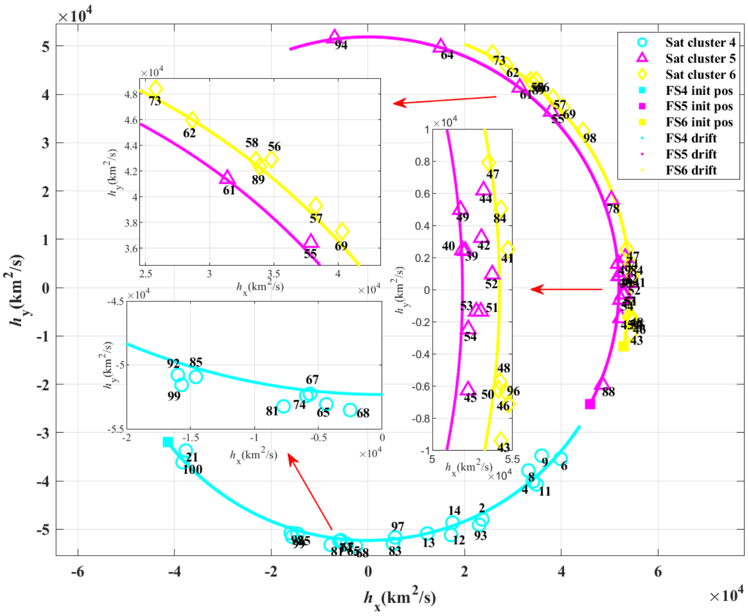
Figure 5.18: Optimal value tuning of λ (100 CSs and 6 SSs scenario).

Table 5.6: Results of CS clustering and optimal orbit design of corresponding SSs (100 CSs and 6 SSs scenario).

SS	$\dot{\Omega}_s^i$ (°/day)	a_s^i (km)	i_s^i (°)	Ω_s^i (°)	$\Delta\Omega_i$ (°)	CS Cluster	Number of CSs	Estimated Fuel Consumption (kg)
s_1	1.15	7087.89	99.59	140.59	113.05	{63,79,60,59,95,82, 30,80,37,32,26,34, 87,71,31,38}	16	681.02
s_2	0.76	9228.50	106.23	56.02	154.93	{15,7,5,10,3,20,25, 24,17,16,19,23,22, 18,28}	15	933.34
s_3	1.23	7609.88	103.25	196.81	171.58	{35,75,33,27,36,29, 90,91,86,72,70,76, 77,66,1}	15	843.32
s_4	1.12	7046.00	99.19	307.67	96.53	{21,100,92,99,85,81, 74,67,65,68,83,97, 13,12,14,93,2,4,11, 8,9,6}	22	1215.50
s_5	1.16	6915.70	98.87	62.35	120.13	{88,45,54,53,51,52, 39,40,42,49,44,78, 55,61,64,94}	16	574.74
s_6	1.09	7713.25	102.26	77.06	71.83	{43,46,96,50,48,41, 84,47,98,69,57,56, 89,58,62,73}	16	434.02
Total	/	/	/	/	/	/	100	4681.94



(a) Distribution of cluster C_1-C_3 and the corresponding SSs



(b) Distribution of cluster C_4-C_6 and the corresponding SSs

Figure 5.19: Distribution of CS clusters and orbit drift of SSs (100 CSs and 6 SSs scenario).

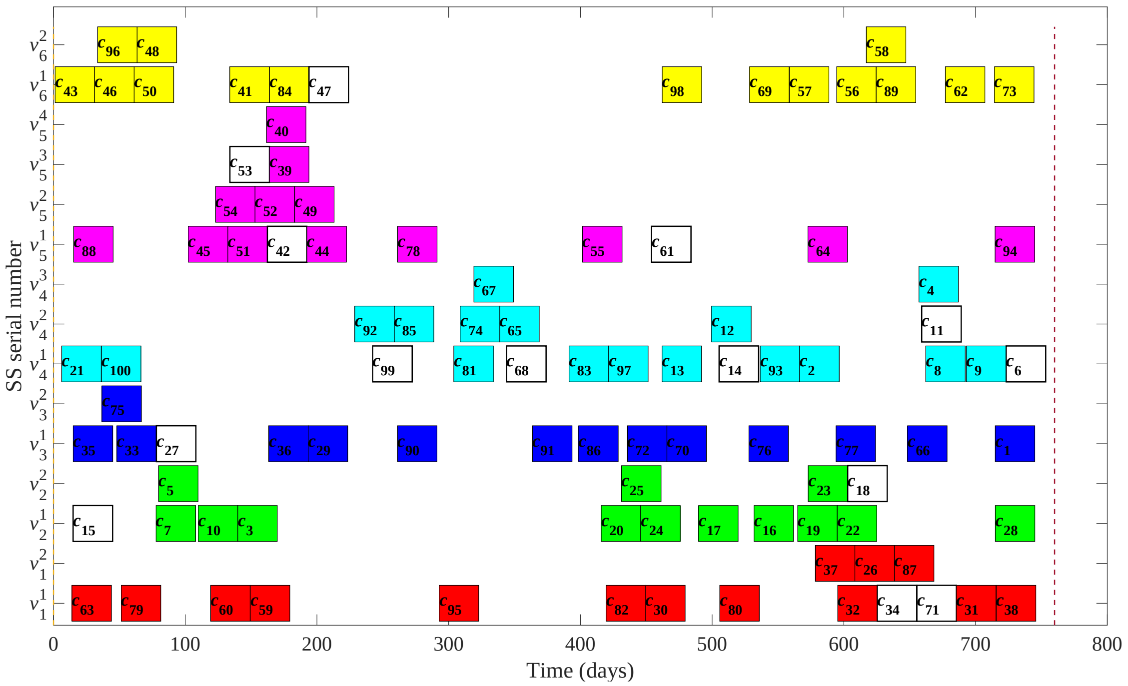


Figure 5.20: OOS mission operation timeline of each service satellite in each SS (100 CSs and 6 SSs scenario).

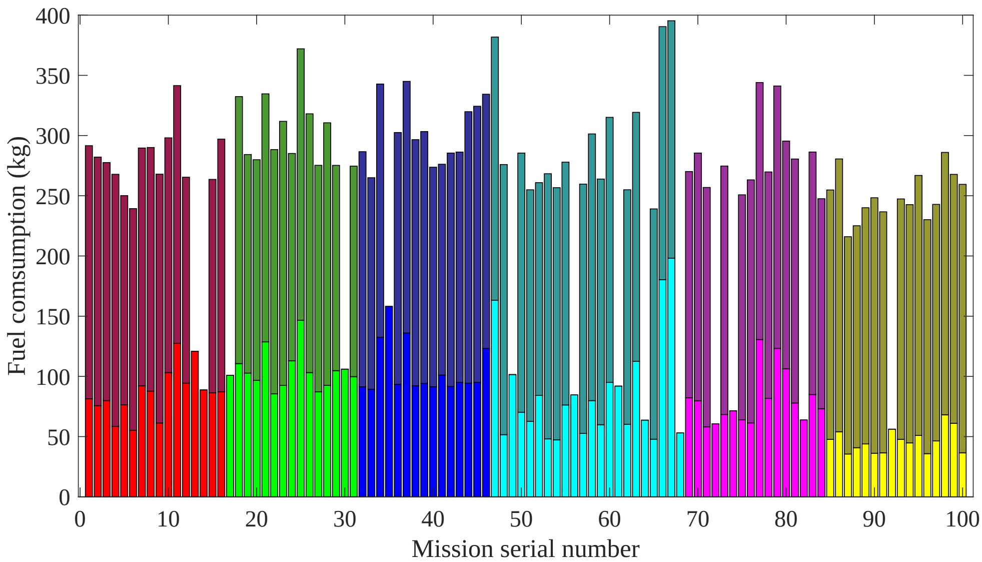


Figure 5.21: Fuel cost for each OOS mission obtained by service satellite mission scheduling (100 CSs and 6 SSs scenario).

Table 5.7: Result comparison of the four algorithms (100 CSs and 6 SSs scenario).

Algorithm	Fitness Value			Total Fuel Consumption (kg)			Number of Service Satellites Used		
	Ave.	Min	Std.	Ave.	Min	Std.	Ave.	Min	Std.
GQA	**5702.92**	**5604.57**	**59.69**	4260.29	4233.73	15.81	**17.10**	**15**	**0.85**
GA[83]	6441.41	6189.11	351.88	**4183.91**	**4170.44**	7.64	19.65	18	0.87
ACO[89]	6122.28	5957.71	134.99	4364.26	4320.12	32.34	18.40	17	0.94
PSO[77]	6417.33	6142.54	153.38	4191.89	4177.12	**7.33**	19.65	18	0.99

Ave.: Average; Min: Minimum; Std.: Standard Deviation.

Table 5.8: Comparison of service satellite mission scheduling results (100 CSs and 6 SSs scenario).

SS	Total Fuel Consumption (kg)					Number of Service Satellites Used				
	GQA	GA[83]	ACO[89]	PSO[77]	GCM[90]	GQA	GA	ACO	PSO	GCM
s_1	641.43	**629.26**	656.04	630.50	732.05	**2**	3	2	3	4
s_2	807.49	**783.07**	810.28	788.14	922.38	**3**	3	3	3	4
s_3	898.09	**868.11**	868.33	868.12	1017.56	**1**	2	2	2	2
s_4	891.51	**877.41**	909.99	878.08	1001.70	**3**	3	3	3	4
s_5	648.60	**644.69**	698.94	652.40	755.88	**4**	5	5	4	6
s_6	398.36	**382.06**	408.48	385.60	440.52	**2**	3	3	3	4
Total	4285.5	**4184.6**	4352.1	4202.8	4870.1	**15**	19	18	18	24

small population size. GQA shows a significant advantage in the key objective of minimizing the number of service satellites, with the best result requiring only 15 service satellites, far fewer than other algorithms. However, GA performs slightly better in terms of total fuel consumption, revealing a trade-off between fuel consumption and the number of service satellites. As seen in Table 5.8, the results of all four optimization algorithms are superior to the GCM method. Notably, GCM requires 6 service satellites for SS s_5, exceeding the maximum limit and highlighting the necessity of optimized scheduling in large-scale scenarios. It is worth noting that the GQA solution's fuel consumption is only about 2.3% higher than the best algorithm, yet it reduces the number of service satellites by 16.7%. This trade-off of a "slight increase in fuel for a significant reduction in service satellites" is more economically valuable in practical applications. Furthermore, GQA's total fuel consumption of 4285.5 kg is lower than the estimated value in Table 5.6, indicating that GQA can effectively utilize the time window characteristics to optimize resource allocation during mission scheduling.

The mission execution timeline generated by GQA, depicted in Figure 5.20, confirms that the schedule for each service satellite is free of time conflicts, with all missions completed before the deadline. Furthermore, as shown in Figure 5.21, the fuel consumption for every OOS mission remains within the service satellite's capacity. These results validate the feasibility of the generated plan and underscore the effectiveness of the proposed framework in handling large-scale, dense mission scenarios.

5.7 Summary

This chapter presents an improved two-stage optimization framework for the mission planning of OOS for LEO constellations under the "service station-service satellite" mode. This framework decouples the problem into two subproblems: the optimal orbit deployment of SSs and the mission scheduling of service satellites, which are solved sequentially.

In the first stage, a Modified Spectral Clustering-Nonlinear Programming (MSC-NLP) method is designed to solve the problem of CS assignment and optimal orbit design for multiple SSs. This stage analyzes two scenarios of relative precession between the SS and the target constellation. A method for calculating the minimum enveloping RAAN angle of a CS cluster is proposed to determine the initial RAAN of the SS. A novel distance metric for CS clustering is also introduced, which significantly reduces the estimated fuel consumption in the SS orbit design.

In the second stage, a Genetic Quantum Algorithm (GQA) is employed to solve the Mixed-Integer Nonlinear Programming (MINLP) model for the service satellite mission scheduling problem. For the first time in this context, the number of service satellites used is included as one of the optimization objectives.

The effectiveness of the proposed framework is validated using the multi-RGT orbit regional coverage constellation case from Chapter 3. A comparison with the results from [90] on a small-scale case demonstrates the superiority of our framework. Furthermore, simulations in a large-scale scenario with 100 CSs confirm the framework's strong applicability to large-scale mission scenarios, showcasing its significant potential for practical applications.

While the proposed GQA demonstrate superior solution quality, their computational complexity poses challenges for ultra-large-scale problems. For strategic deployment and long-term servicing planning, the current runtime is acceptable. However, for real-time mission requirements, the algorithms could be adapted by restricting the search space to local clusters or utilizing the pre-trained results as a warm-start, significantly reducing the convergence time.

Chapter 6

Conclusions and Suggestions for Future Research

This dissertation focuses on the key system optimization problems in the life cycle of LEO satellite constellations, including preliminary design, launch and deployment, and on-orbit servicing. It systematically investigates the associated core system design and optimization problems, with a particular emphasis on innovative approaches to problem modeling and solution algorithms. The effectiveness and superiority of the proposed methods have been validated through extensive simulations. The main research contributions are summarized as follows:

(1) For the design of LEO constellations for continuous regional coverage, a mathematical model named Simultaneous Orbit Optimization and Satellite Deployment (SOOSD) based on multiple RGT orbits was proposed. A two-stage optimization framework was also designed to solve the problem.

This study first investigates the design of LEO constellations for continuous regional coverage using RGT orbits. A deployment model for a single RGT is established by applying the circular convolution property to the constellation pattern and target visibility vectors. Existing models often face limitations such as a restricted solution space or infeasibility, due to a lack of orbital parameter optimization. To overcome these issues, the model is extended into the SOOSD model. This new model co-optimizes orbital parameters and satellite deployment across multiple RGTs, aiming to minimize the number of satellites while meeting coverage requirements. To address the computational challenges of the SOOSD model, which involves high-dimensional decision variables and sparse visibility constraints, a two-stage optimization framework is proposed. This framework combines a differential evolution algorithm with an integer programming solver. It facilitates a step-by-step optimization of both orbital parameters and satellite deployment positions. Simulation results show that this method significantly reduces the required number of satellites compared to traditional symmetric designs and direct solving approaches. Furthermore, it demonstrates strong robustness against variations in the target communication elevation angle.

(2) For the bi-objective optimization problem of constellation L&D using OTVs, a solution framework combining the ϵ-constraint method and a Clustering-based Nearest Neighbor Sorting with Local Search Genetic Algorithm (CNS-LSGA) was proposed.

This dissertation develops a bi-objective combinatorial optimization model that considers both the number of launches and fuel consumption. To handle the problem of a contracted feasible region and the difficulty in obtaining feasible solutions due to resource constraints, such as OTV fuel capacity and rocket payload limits, the CNS-LSGA was designed. This algorithm integrates a clustering and nearest-neighbor sorting initialization mechanism with a local search operator to efficiently generate L&D schemes for a given number of launches. By further combining this with the ϵ-constraint method, the Pareto front of the original bi-objective problem was effectively explored. Extensive simulations and comparative experiments show that the proposed method offers significant advantages in solution quality, robustness, and diversity, providing an efficient tool and a quantitative benchmark for constellation L&D mission planning.

(3) For the LEO constellation OOS mission planning problem, a "Modified Spectral Clustering + Nonlinear Programming" method for service station orbit design was proposed, and a GQA was utilized to co-optimize the number of servicing satellites and fuel consumption.

This dissertation proposes a two-stage optimization framework for the OOS system mission planning of LEO constellations based on "service station-servicing satellite" OOS mode. In the service station orbit design stage, a Modified Spectral Clustering and Nonlinear Programming (MSC-NLP) method is employed to achieve target satellite allocation and optimal orbit design for multiple service stations. By developing a method to calculate the minimum RAAN envelope for targets and introducing a clustering distance metric that considers the orbital distribution of targets, the estimated fuel consumption for servicing satellite missions was effectively reduced. In the servicing satellite mission scheduling stage, a Quantum Genetic Algorithm (QGA) is used to efficiently solve the MINLP model for mission scheduling. This achieves a synergistic optimization of the number of servicing satellites used and their total fuel consumption. Simulation results indicate that this framework not only outperforms existing methods but also demonstrates good applicability and engineering potential in large-scale mission scenarios.

In addition to the theoretical and algorithmic contributions, the practical implementation of the proposed constellation design and servicing frameworks requires careful consideration of real-world engineering constraints. Future research should further address the challenges of orbit control accuracy, where perturbation forces and sensor noise may affect the precision of satellite positioning and fuel consumption estimates. Furthermore, maintaining stable communication links is crucial for multi-satellite coordination during complex on-orbit servicing missions. The stability of these links can be impacted by dynamic geometry and signal interference. Incorporating robust control strategies and decentralized coordination protocols would be essential to bridge the gap between optimized theoretical models and the high-reliability requirements of actual satellite operations. Therefore, future research can be extended in the following directions:

(1) For coverage constellation design, other practical factors such as system redundancy, L&D costs, and communication performance are also crucial. Future work should integrate these factors into the existing design model to obtain results that better align with practical requirements.

(2) For L&D optimization, considering changes in the injection orbit resulting from variations in payload configuration is necessary. Subsequent research should include injection altitude, launch site, and launch vehicle type selection as decision variables to better balance economy and timeliness in L&D planning.

(3) For OOS mission planning, future efforts should aim to extend them to dynamic mission scenarios. This includes addressing unexpected failures in servicing satellites or targets, as well as adapting to evolving mission requirements over time. Developing adaptive scheduling algorithms and real-time decision-making frameworks will be essential to enhance the resilience and flexibility of OOS operations in LEO constellations.

Lastly, future efforts should also focus on enhancing the computational efficiency of all the optimization frameworks mentioned above. Potential acceleration strategies include implementing parallel computing architectures to handle population evaluations and developing surrogate-based models to approximate high-fidelity orbital simulations. These enhancements will facilitate the transition from offline optimization to near-real-time mission planning, enabling rapid responses to various complex dynamic demands.

Bibliography

[1] C. G. Green, P. Massatt, and N. Rhodus, “The GPS 21 primary satellite constellation,” *Navigation*, vol. 36, no. 1, pp. 9–24, 1989.

[2] P. Steigenberger and O. Montenbruck, “Galileo status: Orbits, clocks, and positioning,” *GPS solutions*, vol. 21, no. 2, pp. 319–331, 2017.

[3] Y. Yang, Y. Mao, and B. Sun, “Basic performance and future developments of beidou global navigation satellite system,” *Satellite Navigation*, vol. 1, no. 1, p. 1, 2020.

[4] C. E. Fossa, R. A. Raines, G. H. Gunsch, and M. A. Temple, “An overview of the IRIDIUM (R) low Earth orbit (LEO) satellite system,” in *Proceedings of the IEEE 1998 National Aerospace and Electronics Conference. NAECON 1998. Celebrating 50 Years (Cat. No. 98CH36185)*, IEEE, Dayton, OH, USA, 1998, pp. 152–159.

[5] F. J. Dietrich, P. Metzen, and P. Monte, “The Globalstar cellular satellite system,” *IEEE Transactions on Antennas and Propagation*, vol. 46, no. 6, pp. 935–942, 1998.

[6] S. Cakaj, "The parameters comparison of the "Starlink" LEO satellites constellation for different orbital shells," *Frontiers in Communications and Networks*, vol. 2, p. 643 095, 2021.

[7] S. Kozhaya and Z. M. Kassas, "A first look at the OneWeb LEO constellation: Beacons, beams, and positioning," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 60, no. 5, pp. 7528–7534, 2024.

[8] Y. Yu, H. Jiang, J. Cai, and Q. Zheng, "Analyses of qianfan constellation coverage with respect to the ground users as of may 23, 2025," in *Journal of Physics: Conference Series*, IOP Publishing, vol. 3073, 2025, p. 012 006.

[9] J. Anger, C. de Franchis, and G. Facciolo, "Assessing the sharpness of satellite images: Study of the planetscope constellation," in *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*, IEEE, 2019, pp. 389–392.

[10] D. Li, M. Wang, and J. Jiang, "China's high-resolution optical remote sensing satellites and their mapping applications," *Geo-spatial Information Science*, vol. 24, no. 1, pp. 85–94, 2021.

[11] T. Sung and J. Ahn, "Optimal deployment of satellite mega-constellation," *Acta Astronautica*, vol. 202, pp. 653–669, 2023.

[12] S. Cornara, T. Beech, M. Belló-Mora, and A. Martinez de Aragon, "Satellite constellation launch, deployment, replacement and end-of-life strategies," in *Proceedings of the Small Satellite Conference*, Logan, Utah, United States, 1999.

[13] F. Barato, E. Toson, F. Milza, and D. Pavarin, "Investigation of different strategies for access to space of small satellites on a defined LEO orbit," *Acta Astronautica*, vol. 222, pp. 11–28, 2024.

[14] C. Ward, A. D. S. Curiel, M. Sweeting, G. Aglietti, and J. Schaffner, "Surrey small satellite transfer vehicle," in *International Astronautical Federation - 56th International Astronautical Congress*, vol. 4, Fukuoka, Japan, 2005, pp. 2169–2178.

[15] R. Chang *et al.*, "Satcarrier conops: A concept of operations for satellite carriers," in *7th Responsive Space Conference*, Citeseer, Los Angeles, CA, 2009.

[16] J. Andrews, C. Blake, and M. Wuerl, "Spaceflight secondary payload system (SSPS) and sherpa tug - Enabling systems for the launch of secondary and hosted payloads," in *Proceedings of the International Astronautical Congress, IAC*, vol. 6, Naples, Italy, 2012, pp. 4437–4444.

[17] C. Li and B. Xu, "Optimal scheduling of multiple sun-synchronous orbit satellites refueling," *Advances in Space Research*, vol. 66, no. 2, pp. 345–358, 2020.

[18] M. Pyrak and J. Anderson, "Performance of northrop grumman's mission extension vehicle (MEV) RPO imagers at GEO," in *Autonomous Systems: Sensors, Processing and Security for Ground, Air, Sea and Space Vehicles and Infrastructure 2022*, SPIE, vol. 12115, Orlando, Florida, United States, 2022, pp. 64–82.

[19] A. Kumar, "Emerging commercial services and capabilities: Advancements in on-orbit satellite servicing," in *AIAA AVIATION FORUM AND ASCEND 2025*, 2025, p. 4107.

[20] K. Yang, "Chinese perspective on an international regime of space traffic management," *Journal of Space Safety Engineering*, vol. 6, no. 2, pp. 156–160, 2019.

[21] X. Li, D. Yang, and H. Liu, "China's space robotics for on-orbit servicing: The state of the art," *National Science Review*, vol. 10, no. 5, nwac129, 2023.

[22] Z. Wang, P. Wang, J. Duan, and W. Tian, "Review of on-orbit assembly technology with space robots," *Aerospace*, vol. 12, no. 5, p. 375, 2025.

[23] T. Imaida, Y. Yokokohji, T. Doi, M. Oda, and T. Yoshikawa, "Ground-space bilateral teleoperation of ETS-VII robot arm by direct bilateral coupling under 7-s time delay condition," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 3, pp. 499–511, 2004.

[24] R. B. Friend, "Orbital express program summary and mission overview," in *Sensors and Systems for Space Applications II*, SPIE, vol. 6958, Orlando, Florida, United States, 2008, pp. 11–21.

[25] F. Sheet, "Geosynchronous space situational awareness program," *US Air Force, Air Force Space Command (Archived), current as of September*, 2019.

[26] J. Bultitude, Z. Burkhardt, M. Harris, M. Jelderda, S. A. Suresh, L. Fettes, D. Faber, J. Schiel, J. Cho, D. Levitt, *et al.*, "Development and launch of the world's first orbital propellant tanker," in *Proceedings of the Small Satellite Conference*, Logan, Utah, United States, 2021.

[27] C. Dai, M. Zhang, C. Li, J. Zhao, and Q. Chen, "QoE-aware intelligent satellite constellation design in satellite Internet of Things," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4855–4867, 2021.

[28] H. W. Lee, S. Shimizu, S. Yoshikawa, and K. Ho, "Satellite constellation pattern optimization for complex regional coverage," *Journal of Spacecraft and Rockets*, vol. 57, no. 6, pp. 1309–1327, 2020.

[29] S. S. Lee, “Target-oriented satellite constellation method for revisit performance,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–11, 2023.

[30] F. Ma, X. Zhang, X. Li, J. Cheng, F. Guo, J. Hu, and L. Pan, “Hybrid constellation design using a genetic algorithm for a LEO-based navigation augmentation system,” *GPS Solutions*, vol. 24, no. 2, p. 62, 2020.

[31] G. M. Capez, S. Henn, J. A. Fraire, and R. Garello, “Sparse satellite constellation design for global and regional direct-to-satellite IoT services,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 58, no. 5, pp. 3786–3801, 2022.

[32] Y. Imoto, S. Satoh, T. Obata, and K. Yamada, “Optimal constellation design based on satellite ground tracks for Earth observation missions,” *Acta Astronautica*, vol. 207, pp. 1–9, 2023.

[33] M. Zhou, Z. Song, G. Dai, M. Wang, and X. Chen, “Characteristic area-based method for continuous coverage analysis of satellites constellation,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 59, no. 5, pp. 7128–7139, 2023.

[34] S. Huang, C. Colombo, and F. Bernelli-Zazzera, “Multi-criteria design of continuous global coverage Walker and Street-of-Coverage constellations through property assessment,” *Acta Astronautica*, vol. 188, pp. 151–170, 2021.

[35] Y. Ulybyshev, “General analysis method for discontinuous coverage satellite constellations,” *Journal of Guidance, Control, and Dynamics*, vol. 38, no. 12, pp. 2475–2483, 2015.

[36] X. He and H. Li, “Analytical solutions for Earth discontinuous coverage of satellite constellation with repeating ground tracks,” *Chinese Journal of Aeronautics*, vol. 35, no. 10, pp. 275–291, 2022.

[37] J. E. Draim, “The breakwell memorial lecture: Satellite constellations,” in *55th International Astronautical Congress of the International Astronautical Federation, the International Academy of Astronautics, and the International Institute of Space Law*, Vancouver, BC, Canada, 2004, A–5.

[38] A. Kak and I. F. Akyildiz, “Designing large-scale constellations for the internet of space things with cubesats,” *IEEE Internet of Things Journal*, vol. 8, no. 3, pp. 1749–1768, 2020.

[39] L. Rider, “Analytic design of satellite constellations for zonal Earth coverage using inclined circular orbits,” *Journal of the Astronautical Sciences*, vol. 34, pp. 31–64, 1986.

[40] A. H. Ballard, “Rosette constellations of Earth satellites,” *IEEE Transactions on Aerospace and Electronic Systems*, no. 5, pp. 656–673, 1980.

[41] E. Frayssinhes and E. Lansard, “Optimization of small satellite constellations in low Earth orbit,” *Small Satellites System and Services, Biarritz, CEPADUES ed*, 1994.

[42] E. Frayssinhes, P. Janniere, and E. Lansard, “The use of genetic algorithms in the optimization of satellite constellations,” *Spaceflight Dynamics*, pp. 971–982, 1995.

[43] E. Frayssinhes, “Investigating new satellite constellation geometries with genetic algorithms,” in *AAS/AIAA Astrodynamics Conference*, San Diego, California, 1996, p. 3636.

[44] G. Baselt, M. Strohmeier, J. Pavur, V. Lenders, and I. Martinovic, “Security and privacy issues of satellite communication in the avlatlon domain,” in *2022 14th*

International Conference on Cyber Conflict: Keep Moving! (CyCon), vol. 700, Tallinn, Estonia, 2022, pp. 285–307.

[45] A. Venkatesan, J. Lowenthal, P. Prem, and M. Vidaurri, “The impact of satellite constellations on space as an ancestral global commons,” *Nature Astronomy*, vol. 4, no. 11, pp. 1043–1048, 2020.

[46] H. W. Lee, P. C. Jakob, K. Ho, S. Shimizu, and S. Yoshikawa, “Optimization of satellite constellation deployment strategy considering uncertain areas of interest,” *Acta Astronautica*, vol. 153, pp. 213–228, 2018.

[47] X. Dong, Y. Sui, Y. Li, Z. Chen, and C. Hu, “Repeat ground track SAR constellation design using revisit time image extrapolation and lookup-table-based optimization,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–13, 2023.

[48] P. Chadalavada and A. Dutta, “Regional cubesat constellation design to monitor hurricanes,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–8, 2022.

[49] S. S. Lee, “Satellite constellation method for ground targeting optimized with k-means clustering and genetic algorithm,” *Engineering Applications of Artificial Intelligence*, vol. 129, p. 107 509, 2024.

[50] Y. Ulybyshev, “Satellite constellation design for complex coverage,” *Journal of Spacecraft and Rockets*, vol. 45, no. 4, pp. 843–849, 2008.

[51] N. Mailhot and P. Gurfil, “Constellation design for regional space-borne geolocation,” *Journal of Guidance, Control, and Dynamics*, vol. 45, no. 5, pp. 795–814, 2022.

[52] X. Wang, H. Zhang, S. Bai, and Y. Yue, “Design of agile satellite constellation based on hybrid-resampling particle swarm optimization method,” *Acta Astronautica*, vol. 178, pp. 595–605, 2021.

[53] C. Circi, E. Ortore, and F. Bunkheila, “Satellite constellations in sliding ground track orbits,” *Aerospace Science and Technology*, vol. 39, pp. 395–402, 2014.

[54] E. Ortore, M. Cinelli, and C. Circi, “A ground track-based approach to design satellite constellations,” *Aerospace Science and Technology*, vol. 69, pp. 458–464, 2017.

[55] X. He, H. Li, J. Zhao, and H. Cui, “Configuration design of hybrid constellations with synchronous orbital precession based on rapid coverage analysis,” *Advances in Space Research*, vol. 70, no. 10, pp. 3065–3077, 2022.

[56] H. W. Lee and K. Ho, “Regional constellation reconfiguration problem: Integer linear programming formulation and lagrangian heuristic method,” *Journal of Spacecraft and Rockets*, vol. 60, no. 6, pp. 1828–1845, 2023.

[57] S. S. Lee, “Closed-form solution of repeat ground track orbit design and constellation deployment strategy,” *Acta Astronautica*, vol. 180, pp. 588–595, 2021.

[58] J. M. Colombi, L. D. Buckle, J. Black, and S. G. Nurre, “Optimal launch manifesting for heterogeneous disaggregated satellite constellations,” *Journal of Spacecraft and Rockets*, vol. 54, no. 3, pp. 582–591, 2017.

[59] G. Di Pasquale, B. I. E. Estrada, D. González-Arribas, M. Sanjurjo-Rivo, and D. P. Grande, “Multi-disciplinary optimization of constellation deployment strategies including launcher selection,” *Acta Astronautica*, vol. 223, pp. 389–403, 2024.

[60] M. Bakhtiari and E. Abbasali, “An optimal deployment strategy for multi-plane satellite constellation using a generalized non-planar maneuver,” *Astrophysics and Space Science*, vol. 369, no. 3, p. 26, 2024.

[61] H. Leppinen, “Deploying a single-launch nanosatellite constellation to several orbital planes using drag maneuvers,” *Acta Astronautica*, vol. 121, pp. 23–28, 2016.

[62] G. Di Pasquale, M. Sanjurjo-Rivo, and D. Pérez Grande, “Optimization of constellation deployment using on-board propulsion and Earth nodal regression,” *Advances in Space Research*, vol. 70, no. 11, pp. 3281–3300, 2022.

[63] S. Liu, T. Meng, J. Lei, and Z. Jin, “Kuhn munkres matching deployment optimization method for micro satellite constellation,” *Journal of Astronautics*, vol. 42, no. 7, p. 895, 2021.

[64] W. Xue, Y. Rusn, M. Hu, and W. Guo, “Deployment optimization method of large-scale LEO constellation based on nsga-ii,” in *Proceedings of the International Astronautical Congress, IAC*, Baku, Azerbaijan, 2023.

[65] D. Roychowdhury, S. Grau, and E. Stoll, “Helix orbit deployment of nanosatellites for formation flight: Strategy, gnss/tles evaluation, and accuracy analysis,” *Acta Astronautica*, vol. 232, pp. 244–257, 2025.

[66] J.-C. Zhang, Y. Zhang, Y.-H. Zhu, and Y.-Z. Luo, “Integrated packing modeling for efficient optimization of multispacecraft successive rendezvous,” *Acta Astronautica*, 2025.

[67] A. L. Rivera and M. Naeije, “Neural combinatorial optimization for multi-rendezvous mission design,” *Advances in Space Research*, 2025.

[68] J. Zhang, G. T. Parks, Y. Luo, and G. Tang, "Multispacecraft refueling optimization considering the J2 perturbation and window constraints," *Journal of Guidance, Control, and Dynamics*, vol. 37, no. 1, pp. 111–122, 2014.

[69] A.-Y. Huang, H.-N. Li, J.-C. Zhang, and Y.-Z. Luo, "Orbit traveling salesman problems: New modeling formulation via convexification approach," *Journal of Guidance, Control, and Dynamics*, vol. 47, no. 9, pp. 2008–2015, 2024.

[70] Y. Zhou, Y. Yan, X. Huang, and L. Kong, "Optimal scheduling of multiple geosynchronous satellites refueling based on a hybrid particle swarm optimizer," *Aerospace Science and Technology*, vol. 47, pp. 125–134, 2015.

[71] A. Baniamerian, M. Bashiri, and R. Tavakkoli-Moghaddam, "Modified variable neighborhood search and genetic algorithm for profitable heterogeneous vehicle routing problem with cross-docking," *Applied Soft Computing*, vol. 75, pp. 441–460, 2019.

[72] H. Xu, B. Song, Y. Guo, L. Liu, X. Li, and G. Ma, "An optimization framework with improved auction-based initialization for highly constrained on-orbit servicing mission planning," *Applied Soft Computing*, vol. 149, p. 110 983, 2023.

[73] J. P. Davis, J. P. Mayberry, and J. P. Penn, "On-orbit servicing: Inspection repair refuel upgrade and assembly of satellites in space," *The Aerospace Corporation, report*, vol. 25, 2019.

[74] H. Yan, L. Yuan, C. Li, Y. Guo, G. Ran, and B. Wang, "Rapid on-orbit repair mission planning for multiple geo spacecraft based on a mixed repair strategy," *IEEE Transactions on Aerospace and Electronic Systems*, 2024.

[75] Y. Zhou, Y. Yan, X. Huang, Y. Yang, and H. Zhang, “Mission planning optimization for the visual inspection of multiple geosynchronous satellites,” *Engineering Optimization*, vol. 47, no. 11, pp. 1543–1563, 2015.

[76] K.-M. Wang, J. Zhang, X. Sun, and L.-T. Gao, “Multi-satellite flyby trajectory planning for uncertain target search,” *Acta Astronautica*, vol. 228, pp. 58–68, 2025.

[77] Y. Zhou, Y. Yan, X. Huang, and L. Kong, “Mission planning optimization for multiple geosynchronous satellites refueling,” *Advances in Space Research*, vol. 56, no. 11, pp. 2612–2625, 2015.

[78] H.-X. Shen, T.-J. Zhang, L. Casalino, and D. Pastrone, “Optimization of active debris removal missions with multiple targets,” *Journal of Spacecraft and Rockets*, vol. 55, no. 1, pp. 181–189, 2018.

[79] Z. Wei, T. Long, R. Shi, Y. Wu, and N. Ye, “Scheduling optimization of multiple hybrid-propulsive spacecraft for geostationary space debris removal missions,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 58, no. 3, pp. 2304–2326, 2021.

[80] N. Zhang, S. Chen, Z. Zhang, and H. Baoyin, “Two-stage dynamic-assignment optimization method for multispacecraft debris removal,” *Journal of Guidance, Control, and Dynamics*, vol. 45, no. 9, pp. 1750–1759, 2022.

[81] C. Han, S. Zhang, and X. Wang, “On-orbit servicing of geosynchronous satellites based on low-thrust transfers considering perturbations,” *Acta Astronautica*, vol. 159, pp. 658–675, 2019.

[82] T. Sarton du Jonchay, H. Chen, O. Gunasekara, and K. Ho, “Rolling horizon optimization framework for the scheduling of on-orbit servicing operations under servicing demand uncertainties,” in *ASCEND 2020*, Virtual Event, 2020, p. 4131.

[83] Z. Zhao, J. Zhang, H.-y. Li, and J.-y. Zhou, “LEO cooperative multi-spacecraft refueling mission optimization considering j2 perturbation and target’s surplus propellant constraint,” *Advances in Space Research*, vol. 59, no. 1, pp. 252–262, 2017.

[84] J. Yu, X.-Q. Chen, L.-H. Chen, and D. Hao, “Optimal scheduling of GEO debris removing based on hybrid optimal control theory,” *Acta Astronautica*, vol. 93, pp. 400–409, 2014.

[85] B. Yan, J. Zhang, Y.-Z. Luo, and Y.-H. Zhu, “Multitarget flyby assignment and sequence planning with customized ant colony optimization,” *Journal of Guidance, Control, and Dynamics*, vol. 47, no. 2, pp. 375–383, 2024.

[86] H. Shen and P. Tsiotras, “Peer-to-peer refueling for circular satellite constellations,” *Journal of Guidance, Control, and Dynamics*, vol. 28, no. 6, pp. 1220–1230, 2005.

[87] A. Dutta and P. Tsiotras, “Network flow formulation for cooperative peer-to-peer refueling strategies,” *Journal of Guidance, Control, and Dynamics*, vol. 33, no. 5, pp. 1539–1549, 2010.

[88] A. Dutta, N. Arora, and R. P. Russell, “Peer-to-peer refueling strategy using low-thrust propulsion,” *Journal of Spacecraft and Rockets*, vol. 49, no. 5, pp. 944–954, 2012.

[89] T.-J. Zhang, Y.-K. Yang, B.-H. Wang, Z. Li, H.-X. Shen, and H.-N. Li, "Optimal scheduling for location geosynchronous satellites refueling problem," *Acta Astronautica*, vol. 163, pp. 264–271, 2019.

[90] X. Zhu, C. Zhang, R. Sun, J. Chen, and X. Wan, "Orbit determination for fuel station in multiple sso spacecraft refueling considering the J2 perturbation," *Aerospace Science and Technology*, vol. 105, p. 105 994, 2020.

[91] M. A. Luu and D. E. Hastings, "On-orbit servicing system architectures for proliferated low-earth-orbit constellations," *Journal of Spacecraft and Rockets*, vol. 59, no. 6, pp. 1946–1965, 2022.

[92] X.-Q. Chen and J. Yu, "Optimal mission planning of geo on-orbit refueling in mixed strategy," *Acta Astronautica*, vol. 133, pp. 63–72, 2017.

[93] B. Meng, J. Huang, Z. Li, L. Huang, Y. Pang, X. Han, and Z. Zhang, "The orbit deployment strategy of oos system for refueling near-Earth orbit satellites," *Acta Astronautica*, vol. 159, pp. 486–498, 2019.

[94] R. Apa, I. Kaminer, J. Hudson, and M. Romano, "Optimal low-thrust orbital transfer for servicing multiple satellites in elliptical orbits," *Acta Astronautica*, vol. 228, pp. 686–699, 2025.

[95] B. Du, Y. Zhao, A. Dutta, J. Yu, and X. Chen, "Optimal scheduling of multispacecraft refueling based on cooperative maneuver," *Advances in Space Research*, vol. 55, no. 12, pp. 2808–2819, 2015.

[96] Y. Zhou, Y. Yan, X. Huang, and Y. Yang, "Multi-objective planning of a multiple geostationary spacecraft refuelling mission," *Engineering Optimization*, vol. 49, no. 3, pp. 531–548, 2017.

[97] S. E. Sorenson and S. G. N. Pinkley, "Multi-orbit routing and scheduling of refuellable on-orbit servicing space robots," *Computers & Industrial Engineering*, vol. 176, p. 108 852, 2023.

[98] H. Li and H. Baoyin, "Optimization of multiple debris removal missions using an evolving elitist club algorithm," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 56, no. 1, pp. 773–784, 2019.

[99] N. Zhang, Z. Zhang, and H. Baoyin, "Timeline club: An optimization algorithm for solving multiple debris removal missions of the time-dependent traveling salesman problem model," *Astrodynamics*, pp. 1–16, 2022.

[100] J. Bang and J. Ahn, "Multitarget rendezvous for active debris removal using multiple spacecraft," *Journal of Spacecraft and Rockets*, vol. 56, no. 4, pp. 1237–1247, 2019.

[101] S. Chen, F. Jiang, H. Li, and H. Baoyin, "Optimization for multitarget, multispacecraft impulsive rendezvous considering J2 perturbation," *Journal of Guidance, Control, and Dynamics*, vol. 44, no. 10, pp. 1811–1822, 2021.

[102] P. Anastassios, G. Daniel, J. Drew, L. Gregory, N. Austin, R. Javier, S. Juan, S. Jeffrey, A. Nitin, P. Thomas, *et al.*, "GTOC9: Results from the jet propulsion laboratory (team JPL)," *Acta Futura*, vol. 11, pp. 25–35, 2018.

[103] G. Jia, P. Zhaojun, and D. Zhonghua, "Optimal planning for a multi-debris active removal mission with a partial debris capture strategy," *Chinese Journal of Aeronautics*, vol. 36, no. 6, pp. 256–265, 2023.

[104] Y.-H. Zhu, Y. Zhang, Y.-Z. Luo, and J.-C. Zhang, "Intelligence-based multistage optimization framework for multiple debris removal mission," *Journal of Guidance, Control, and Dynamics*, vol. 47, no. 7, pp. 1447–1456, 2024.

[105] W. Wu, H. Lu, J. Xu, and Y. Sun, "Hierarchical optimization method of space patrol task assignment based on deep neural network and consensus-based bundle algorithm," *Acta Astronautica*, vol. 207, pp. 295–306, 2023.

[106] H.-L. Choi, L. Brunet, and J. P. How, "Consensus-based decentralized auctions for robust task allocation," *IEEE Transactions on Robotics*, vol. 25, no. 4, pp. 912–926, 2009.

[107] K. Daneshjou, A. Mohammadi-Dehabadi, and M. Bakhtiari, "Mission planning for on-orbit servicing through multiple servicing satellites: A new approach," *Advances in Space Research*, vol. 60, no. 6, pp. 1148–1162, 2017.

[108] X. Zhu, J. Chen, C. Zhang, and B. Qiao, "Optimal fuel station arrangement for multiple geo spacecraft refueling mission," *Advances in Space Research*, vol. 66, no. 8, pp. 1924–1936, 2020.

[109] H. D. Curtis, *Orbital Mechanics for engineering students*. Daytona Beach, Florida: Butterworth-Heinemann, 2013.

[110] D. A. Vallado, *Fundamentals of astrodynamics and applications*. EI Segundo, California: Springer Science & Business Media, 2001, vol. 12.

[111] B. Al Homssi and A. Al-Hourani, "Optimal beamwidth and altitude for maximal uplink coverage in satellite networks," *IEEE Wireless Communications Letters*, vol. 11, no. 4, pp. 771–775, 2022.

[112] M. E. Avendaño, J. J. Davis, and D. Mortari, “The 2-D lattice theory of flower constellations,” *Celestial Mechanics and Dynamical Astronomy*, vol. 116, no. 4, pp. 325–337, 2013.

[113] P. Han, C. Li, C. Huang, H. Huang, Y. Guo, and G. Pan, “Leo internet satellite constellation design for regional civil aviation airways with multiple repeat ground tracks,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 61, no. 2, pp. 4088–4104, 2025.

[114] D. Tian and Z. Shi, “MPSO: Modified particle swarm optimization and its applications,” *Swarm and Evolutionary Computation*, vol. 41, pp. 49–68, 2018.

[115] A. Munawar, M. Wahib, M. Munetomo, and K. Akama, “Advanced genetic algorithm to solve MINLP problems over GPU,” in *2011 IEEE Congress of Evolutionary Computation*, IEEE, New Orleans, Louisiana, USA, 2011, pp. 318–325.

[116] Y.-C. Lin, K.-S. Hwang, and F.-S. Wang, “Co-evolutionary hybrid differential evolution for mixed-integer optimization problems,” *Engineering Optimization*, vol. 33, no. 6, pp. 663–682, 2001.

[117] R. Storn and K. Price, “Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces,” *Journal of Global Optimization*, vol. 11, pp. 341–359, 1997.

[118] N. Ye, T. Long, R. Shi, and Y. Wu, “Radial basis function-assisted adaptive differential evolution using cooperative dual-phase sampling for high-dimensional expensive optimization problems,” *Structural and Multidisciplinary Optimization*, vol. 65, no. 9, p. 241, 2022.

[119] R. Tanabe and A. Fukunaga, "Success-history based parameter adaptation for differential evolution," in *2013 IEEE Congress on Evolutionary Computation*, IEEE, Cancun, Mexico, 2013, pp. 71–78.

[120] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, "PlatEMO: A MATLAB platform for evolutionary multi-objective optimization [educational forum]," *IEEE Computational Intelligence Magazine*, vol. 12, no. 4, pp. 73–87, 2017.

[121] J.-F. Bérubé, M. Gendreau, and J.-Y. Potvin, "An exact ϵ-constraint method for bi-objective combinatorial optimization problems: Application to the traveling salesman problem with profits," *European Journal of Operational Research*, vol. 194, no. 1, pp. 39–50, 2009.

[122] S. Chen and H. Baoyin, "Analytical estimation of the velocity increment in J2-perturbed impulsive transfers," *Journal of Guidance, Control, and Dynamics*, vol. 45, no. 2, pp. 310–319, 2022.

[123] X. Liang, N. Wang, M. Zhang, and B. Jiang, "Bi-objective multi-period vehicle routing for perishable goods delivery considering customer satisfaction," *Expert Systems with Applications*, vol. 220, p. 119 712, 2023.

[124] P. HAN, C. HUANG, C. LI, and Y. GUO, "Bi-objective optimization of orbital transfer vehicle based launch and deployment process for leo constellation," *Chinese Journal of Aeronautics*, p. 103 974, 2025, ISSN: 1000-9361.

[125] K. Krishna and M. N. Murty, "Genetic K-means algorithm," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 29, no. 3, pp. 433–439, 1999.

[126] B. L. Miller, D. E. Goldberg, *et al.*, "Genetic algorithms, tournament selection, and the effects of noise," *Complex systems*, vol. 9, no. 3, pp. 193–212, 1995.

[127] A. Arram and M. Ayob, "A novel multi-parent order crossover in genetic algorithm for combinatorial optimization problems," *Computers & Industrial Engineering*, vol. 133, pp. 267–274, 2019.

[128] S. I. Suliman, G. Kendall, and I. Musirin, "Optimizing channel allocation in wireless communication using single-swap mutation based heuristic," in *2013 15th International Conference on Advanced Communications Technology*, IEEE, PyeongChang, Korea (South), 2013, pp. 774–778.

[129] R. K. Roy, *Design of experiments using the Taguchi approach: 16 steps to product and process improvement*. New York, US: John Wiley & Sons, 2001.

[130] K. M. Sallam, S. M. Elsayed, R. K. Chakrabortty, and M. J. Ryan, "Improved multi-operator differential evolution algorithm for solving unconstrained problems," in *2020 IEEE Congress on Evolutionary Computation*, IEEE, Glasgow, UK, 2020, pp. 1–8.

[131] P. Han, Y. Guo, C. Li, H. Zhi, and Y. Lv, "Multiple GEO satellites on-orbit repairing mission planning using large neighborhood search-adaptive genetic algorithm," *Advances in Space Research*, vol. 70, no. 2, pp. 286–302, 2022.

[132] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[133] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.

[134] H. Ma, H. Wei, Y. Tian, R. Cheng, and X. Zhang, "A multi-stage evolutionary algorithm for multi-objective optimization with complex constraints," *Information Sciences*, vol. 560, pp. 68–91, 2021.

[135] L. While, P. Hingston, L. Barone, and S. Huband, "A faster algorithm for calculating hypervolume," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 1, pp. 29–38, 2006.

[136] S. Cakaj, "The parameters comparison of the "Starlink" LEO satellites constellation for different orbital shells," *Frontiers in Communications and Networks*, vol. 2, p. 643 095, 2021.

[137] P. Han, Y. Guo, P. Wang, C. Li, and W. Pedrycz, "Optimal orbit design and mission scheduling for sun-synchronous orbit on-orbit refueling system," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 59, no. 5, pp. 4968–4983, 2023.

[138] H.-X. Shen and L. Casalino, "Simple Δv approximation for optimization of debris-to-debris transfers," *Journal of Spacecraft and Rockets*, vol. 58, no. 2, pp. 575–580, 2021.

[139] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007.

[140] L. Hagen and A. B. Kahng, "New spectral methods for ratio cut partitioning and clustering," *IEEE Transactions on Computer-aided Design of Integrated Circuits and Systems*, vol. 11, no. 9, pp. 1074–1085, 1992.

[141] L. E. Peterson, “K-nearest neighbor,” *Scholarpedia*, vol. 4, no. 2, p. 1883, 2009.

[142] H. Lutkepohl, “Handbook of matrices,” *Computational Statistics and Data Analysis*, vol. 2, no. 25, p. 243, 1997.

[143] K.-H. Han and J.-H. Kim, “Genetic quantum algorithm and its application to combinatorial optimization problem,” in *Proceedings of the 2000 Congress on Evolutionary Computation*, IEEE, vol. 2, La Jolla, CA, USA, 2000, pp. 1354–1360.

[144] H. Wang, H. He, and Z. Yang, “Scheduling of agile satellites based on an improved quantum genetic algorithm,” *Journal of Astronautics*, vol. 39, no. 11, p. 1266, 2018.