DESIGN AND OPERATIONS OF SATELLITE CONSTELLATIONS FOR COMPLEX REGIONAL COVERAGE

A Dissertation
Presented to
The Academic Faculty

By

Hang Woon Lee

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
College of Engineering
Daniel Guggenheim School of Aerospace Engineering

Georgia Institute of Technology

August 2022

DESIGN AND OPERATIONS OF SATELLITE CONSTELLATIONS FOR COMPLEX REGIONAL COVERAGE

Thesis committee:

Dr. Koki Ho, Advisor
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Harrison H. M. Kim
Department of Industrial and Enterprise Systems Engineering
University of Illinois at Urbana-Champaign

Dr. Brian C. Gunter
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Shoji Yoshikawa
Advanced Technology R&D Center
Mitsubishi Electric Corporation

Dr. E. Glenn Lightsey
School of Aerospace Engineering
Georgia Institute of Technology

Date approved: July 14, 2022

To my family and Hyeonsuh.

ACKNOWLEDGMENTS

The completion of my dissertation could not have been possible without the support that I have received from many individuals. In particular, I feel deeply indebted to my advisor, Dr. Koki Ho, for his unparalleled support during my Ph.D. study. Also, I am grateful to my dissertation committee members, Dr. Brian Gunter, Dr. Glenn Lightsey, Dr. Harrison Kim, and Dr. Shoji Yoshikawa. I am humbled to have them on my dissertation committee for their intellectual input and guidance.

I sincerely appreciate the generous financial support from the National Science Foundation Graduate Research Fellowship Program (Grants DGE-1650044 and DGE-2039655) and the Mitsubishi Electric Corporation. Their support afforded me to freely pursue the research topic that I am enthusiastic about.

I would like to thank the friends I have met during my graduate study. I extend my special thanks to my colleagues, Hao Chen and Tristan Sarton du Jonchay, for our time together. I also send my big thanks to all of my friends at the Space Systems Optimization Group and Space Systems Design Laboratory for our unforgettable moments together.

I express my heartfelt gratitude and love to my family for their unconditional support. Last but not least, I thank Hyeonsuh for standing by my side for the past six years.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Fueled by recent technological advancements in small and capable satellites, satellite constellations are now shaping the new era of space commercialization creating new forms of services that span from Earth observations to telecommunications and navigation. With the mission objectives becoming increasingly complex, a new paradigm in the design and operations of satellite constellations is necessary to make a system cheaper and more efficient.

This dissertation presents a set of novel mathematical formulations and solution methods that lend themselves to various applications in the design and operations of satellite constellation systems. The second chapter establishes the Access-Pattern-Coverage (APC) decomposition model that relaxes the symmetry and homogeneity assumptions of the classical global-coverage constellation design methods. Based on the model, this dissertation formulates an integer linear programming (ILP) problem that designs an optimal constellation pattern for complex spatiotemporally-varying coverage requirements. The third chapter examines the problem of reconfiguring satellite constellations for efficient adaptive mission planning and presents a novel ILP formulation that combines constellation design and transfer problems that are otherwise considered independent and serial in the state-of-the-art. Furthermore, the third chapter proposes a Lagrangian relaxation-based heuristic method that exploits the assignment problem structure embedded in the integrated design-transfer model. The fourth chapter extends the third chapter by investigating the multi-stage satellite constellation reconfiguration problem and develops two heuristic sequential decision-making methods based on the concepts of myopic policy and the rolling horizon procedure. This dissertation presents several illustrative examples as proofs-of-concept to demonstrate the value of the proposed work.

CHAPTER 1
INTRODUCTION

Satellite constellations enable a variety of applications that are otherwise considered infeasible with monolithic satellite systems. The unique advantage of distributed satellite systems is their ability to deliver incomparable coverage performance both spatially (e.g., wide area coverage) and temporally (e.g., rapid revisit time). Fueled by technological advancements in small and capable satellites, we are witnessing a great diversification of satellite constellation applications. Satellite constellations are not only being conceived for applications in Earth's vicinity (e.g., Earth observations, telecommunications, and positioning services) but also for various applications in deep space (e.g., navigation services and space domain awareness in the cislunar regime).

Accompanying the diversification of satellite constellation applications are the mission objectives that are becoming increasingly complex. Key examples of such complex mission objectives include (1) spatiotemporally-varying coverage of disjointed areas and spot targets, (2) a system of multiple orbital shells (e.g., the Satellite-over-Satellite networks), (3) a federation of heterogeneous satellites with different sizes and capabilities, and (4) adaptive mission planning, to name a few. In the face of the increasing complexity of mission objectives, the state-of-the-art constellation architecture design methods (e.g., Walker-delta constellation patterns) become inefficient due to the assumptions that are made to simplify the problem—the symmetry in satellite distributions, and the homogeneity in orbital characteristics and satellite specifications, and static configurations. Therefore, in order to realize efficient next-generation satellite constellation systems, a new paradigm in design and operations is necessary.

In light of this observation, this dissertation presents a set of novel mathematical formulations and solution methods, which lend themselves to various applications in the design

and operations of satellite constellations. In particular, this dissertation builds upon the classical constellation design theories by relaxing the assumptions of homogeneity, symmetry, and static configurations. This allows mission designers and operators to explore larger design space and potentially discover efficient constellation pattern sets and operational procedures. Several computational studies are conducted to demonstrate the value of the proposed work. Note that throughout this dissertation, we use the term "regional-coverage constellations" or regional constellations, in short, to accentuate *heterogeneity* and *asymmetry* as distinctive design philosophies in contrast to the homogeneity and symmetry of classical global-coverage constellations.

This dissertation comprises three distinct chapters. The following is the overview of these chapters.

Chapter 2 establishes the Access-Pattern-Coverage (APC) decomposition model that relaxes the symmetry and homogeneity assumptions commonly made by the classical global-coverage constellation design methods. To this end, the circular convolution nature of the repeating ground track orbit and common ground track constellation is formalized. This formulation enables a scalable constellation pattern analysis for multiple target areas and with multiple sub-constellations. The formalized circular convolution relationship is first used to derive a baseline constellation pattern design method with the conventional assumption of symmetry. Next, a novel method based on binary integer linear programming is developed, which aims to optimally design a constellation pattern with the minimum number of satellites. This binary integer linear programming method achieves optimal constellation patterns for general problem settings that the baseline method cannot achieve. Five illustrative examples are analyzed to demonstrate the value of the proposed new approach.

Chapter 3 investigates the problem of optimizing a satellite constellation reconfiguration process against two competing mission objectives: (i) the maximization of the total coverage reward and (ii) the minimization of the total cost of the transfer. The decision variables for the reconfiguration process include the design of the new configuration and the

assignment of satellites from one configuration to another. This chapter presents a novel bi-objective integer linear programming formulation that combines constellation design and transfer problems. The formulation lends itself to the use of generic mixed-integer linear programming (MILP) methods such as the branch-and-bound algorithm for the computation of provably-optimal solutions; however, these approaches become computationally prohibitive even for moderately-sized instances. In response to this challenge, this chapter proposes a Lagrangian relaxation-based heuristic method that leverages the assignment problem structure embedded in the problem. The results from the computational experiments attest to the near-optimality of the Lagrangian heuristic solutions and significant improvement in the computational runtime compared to a commercial MILP solver.

Chapter 4 extends Chapter 3 by investigating the multi-stage satellite constellation reconfiguration problem. The goal of the problem is to maximize the total system observation throughput by actively manipulating the orbits and the relative phasing of the constituent satellites. This chapter submits a novel integer linear programming formulation of the problem that is constructed based on the concept of time-expanded networks. To tackle the computational intractability arising due to the combinatorial explosion of the solution space, this chapter proposes two computationally efficient heuristic sequential decision-making methods based on the principles of myopic policy and the rolling horizon procedure. Empirical findings show that these heuristics produce high-quality solutions relative to optimal solutions. Computational experiments are conducted to demonstrate the value of the proposed work.

Figure. 1.1 provides the overview of the scope of this dissertation study.

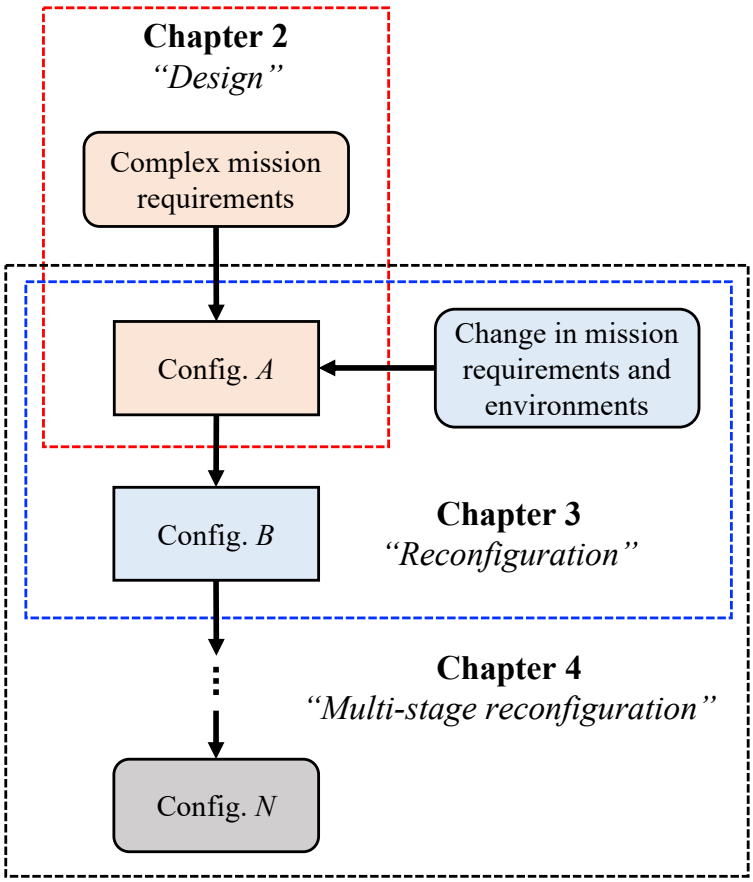


Figure 1.1: Dissertation overview; this dissertation study begins with the idea of "complex regional coverage".

CHAPTER 2
SATELLITE CONSTELLATION PATTERN OPTIMIZATION FOR COMPLEX REGIONAL COVERAGE

2.1 Introduction

Satellite constellations for regional coverage are increasingly being considered as competent business solutions in a market dominated by global-based constellation systems. Regional constellations, whose form varies from being standalone to augmenting existing space-borne systems, provide flexible solutions to stakeholders as a means of circumventing geopolitical, economic, and/or technical issues associated with global constellation systems. Examples of such regional constellation systems are the Indian Regional Navigation Satellite System (IRNSS) [1] and the Quasi-Zenith Satellite System [2].

Unlike global coverage constellations, regional coverage constellations solely focus on the coverage over a local region and therefore generally require a smaller number of satellites in the system to achieve the same performance per area metric compared to global-coverage constellations. This leads to a significantly reduced system cost as the total life-cycle cost of the system depends on the number of satellites [3]. The reduced system cost allows for a tolerable risk of failure and facilitates a shorter payback period. These properties allow regional constellation systems to swiftly react to uncertainties arising from market demand and/or administrative issues. Research has also shown that a flexible option to treat a regional constellation system as part of a larger staged deployment process can be beneficial when market uncertainties are present [4].

Various space systems have been designed for regional coverage. Although the most classical regional coverage method is to use geo-synchronous/-stationary equatorial orbits (GSO/GEO), non-geostationary orbit (NGSO) systems are deemed to provide better per-

formance for many mission-critical attributes such as latency and launch cost. Traditional constellation design methods have investigated the problems with relatively simple coverage criteria, such as satisfying an f-fold continuous coverage requirement (e.g., single-fold, double-fold, etc.) or minimizing the maximum revisit time gap over an area. However, the problems with complex coverage requirements that are periodically time-varying and spatially-varying have not been explored. Examples of such coverage requirements are (1) the increased communication service needs during the daytime; and (2) the increased service needs over urban/sub-urban areas for reliable access [5]. The design process to generate the optimal constellation for such complex coverage requirements involves determining (1) the common orbital characteristics and (2) a constellation pattern. While conventional constellation design methods often assume a symmetric pattern (e.g., Walker constellations [6, 7, 8]) and optimize the common orbital characteristics (e.g., altitude, inclination) with that assumption, the large design space of asymmetric constellation patterns is often missed despite its importance particularly for complex time-varying and spatially-varying coverage requirements. Furthermore, it is reasonable to assume that such a regional coverage constellation system can constitute multiple sub-constellations, each with different orbital characteristics, as demonstrated in the case of IRNSS [1]; however, the concurrent design of multiple sub-constellation patterns using NGSOs requires a sophisticated optimization approach. Such a topic has been scarcely studied and remains an open question. Given this background, a research question of interest arises: "*How do we design a constellation pattern (for multiple sub-constellations if needed) that is optimized (i.e., with the minimum number of satellites) for a periodically time-varying and spatially-varying demand over the regional area(s) of interest?*" This chapter seeks to address this question by constructing an optimal constellation pattern design approach for complex regional coverage. The resulting rigorous constellation pattern design approach can be integrated with existing orbital characteristics design methods and launch/mission constraints to optimize future satellite constellation design.

The contribution of this chapter is as follows. First, the discovery of a *circular convolution phenomenon* between a seed satellite access profile, a constellation pattern vector, and a coverage timeline is formalized in this research. The resulting formulation is referred to as the APC decomposition, following the acronyms of the seed satellite Access profile, constellation Pattern, and Coverage timeline; each of these concepts is introduced in detail later in this chapter. We derive a linear formulation that enables us to design a constellation pattern for a system of multiple sub-constellations for multiple regions. This formulation provides a foundation for general methods introduced herein. Second, we extend the traditional definition of a time-independent f-fold coverage requirement (e.g., single-fold, double-fold, etc.) to a time-dependent $f[n]$-fold coverage requirement, where n is a discrete-time instant, such that periodically time-varying coverage demands can be handled optimally in the constellation design. By applying this idea to multiple target points, this approach is further extended to the case with time-varying and spatially-varying coverage requirements. Finally, we develop a general method based on binary integer linear programming (BILP) that finds the optimal satellite constellation pattern for complex regional coverage, and, if needed, for multiple sub-constellations concurrently. This core concept enables users to explore the hidden design space by breaking the symmetry in the constellation design. The developed constellation pattern design approach is demonstrated with a series of case study examples.

The rest of this chapter is organized as follows. Section 2.2 provides a summary of the key literature relevant to this research. Section 2.3 provides an overview of the constellation model used in this chapter. Section 2.4 introduces the ideas behind the developed approach, including the circular convolution formulation of the problem and its pertinent definitions. Section 2.5 then introduces two methods based on this formulation: the baseline quasi-symmetric and the novel BILP methods. The developed methods are applied to various illustrative examples in Section 2.6 for demonstration. Section 2.7 then concludes this chapter.

2.2 Literature Review

This section reviews the major literature relevant to this study. Traditional satellite constellation design methods have focused on minimizing the number of satellites while providing continuous coverage over a large area of interest such as the globe or latitudinally-bounded zones. Classical methods such as the streets of coverage [9, 10, 11, 12], Walker and Rosette constellations [6, 7, 8, 13], and the tetrahedron elliptical constellation [14] leveraged a geometric approach to exhibit a symmetry in the constellation pattern, where satellites are uniformly and symmetrically arranged based on a predetermined phasing rule. The symmetry in the constellation pattern provides a foundation for a complete design space analysis due to finite variability [15] or for an analytical solution. However, this usually leads to redundant coverage overlaps and therefore may not produce an optimal constellation design in terms of the number of satellites over a bounded local region.

There are several prior studies that specifically dealt with the design of regional coverage constellations. By fully utilizing the characteristics of the repeating ground track orbits, Hanson et al. [16] and Ma and Hsu [17] utilized the timeline meshing method to generate the optimal constellation with respect to minimizing the maximum time gap at the minimum possible inclination. Similarly, Pontani and Teofilatto extended the characteristics of the repeating ground track by searching for allowable time delays with respect to minimizing the gap or maximizing coverage [18]. In addition, Crossley and Williams used metaheuristics methods to design a satellite constellation to minimize the maximum revisit time [19]. Although these regional constellation design algorithms show promising ability to produce asymmetric configuration with respect to a single target point or a connected area, these methods are not applicable to designing a constellation system for periodically time-varying demands over multiple disjoint target points (referred to as complex coverage requirement in this chapter) with multiple sub-constellations. Ulybyshev investigated a new geometric approach to generate satellite constellation designs for complex coverage [20].

The method demonstrates the use of the two-dimensional space and combined maps for the satellite constellation and coverage functions. Nevertheless, this method cannot be applied to asymmetric constellations. Other literature can be found in the comprehensive literature review by Dutruel-Lecohier and Mora as well as Wertz [21, 15]. Recently, Ulybyshev presented a short historical survey of satellite constellation design for continuous coverage [22]. However, there is no methodology that directly answers our question raised in the introduction that considers all three aspects of the regional coverage problem: (1) multiple target points, (2) complex coverage requirements, and (3) multiple sub-constellations.

In response to this background, this chapter attempts to construct methods to design a satellite constellation pattern for periodically time-varying and spatially-varying coverage requirements over multiple target points, and if demanded, for multiple sub-constellations. Building upon the idea of repeating ground track orbits and common ground track constellations (e.g., Flower Constellation set theory [23, 24, 25]) and generalizing our prior work [26], we formalize the circular convolution nature of the constellation pattern design problem and derive two methods for it: (1) the baseline and rather traditional quasi-symmetric method; and (2) the more general and novel BILP method. The developed approach can design the constellation pattern that satisfies the complex coverage requirements of multiple target points with the minimum number of satellites possible exploring both symmetric and asymmetric patterns.

2.3 Satellite Constellation Model

This section introduces the ideas and assumptions on the satellite constellation model that the proposed approach builds upon, including the repeating ground track orbit and the common ground track constellation.

2.3.1 Repeating Ground Track Orbit

A ground track of a satellite is defined as a trace of its sub-satellite points on the surface of the Earth. In this chapter, we utilize a repeating ground track (RGT) orbit as a basis for the orbital design of the constellation, which allows a ground track of a satellite to repeat exactly and periodically. This type of orbit has been shown to provide better coverage performance than the non-repeating ground track orbits with fewer satellites for regional coverage [16]. Considering the Earth-centered Earth-fixed (ECEF) frame, an RGT orbit is achieved when the nodal period of the orbit T_S (the time interval between two consecutive crossings of the orbit ascending node by a satellite) is a rational multiple of the nodal period of Greenwich T_G (the time interval between two consecutive crossings of the orbit ascending node line by the prime meridian):

$$T_r = N_P T_S = N_D T_G \tag{2.1}$$

where T_r represents the period of repetition. Eq. (2.1) implies that a satellite on an RGT orbit makes N_P number of revolutions in N_D number of nodal periods of Greenwich [23, 27]. N_P and N_D are positive integer numbers.

Considering the J_2 perturbation effect, the nodal period of the satellite orbit T_S and the nodal period of Greenwich T_G are given in Eqs. (2.2):

$$T_S = \frac{2\pi}{\dot{\omega} + \dot{M}} \tag{2.2a}$$

$$T_G = \frac{2\pi}{\omega_\oplus - \dot{\Omega}} \tag{2.2b}$$

where $\omega_\oplus$ is the rotation rate of the Earth, $\dot{\omega}$ is the rate of change in the argument of perigee due to perturbations, $\dot{\Omega}$ is the rate of nodal regression of a satellite's orbit, and $\dot{M}$ is the rate of change in the mean anomaly due to nominal motion and perturbations. The perturbed

orbital elements in Eqs. (2.2) are:

$$\dot{\omega} = \frac{3}{2} J_2 \left(\frac{R_\oplus}{p}\right)^2 \sqrt{\frac{\mu_\oplus}{a^3}} \left[2 - \frac{5}{2}\sin^2 i\right] \tag{2.3}$$

$$\dot{\Omega} = -\frac{3}{2} J_2 \left(\frac{R_\oplus}{p}\right)^2 \sqrt{\frac{\mu_\oplus}{a^3}} \cos i \tag{2.4}$$

$$\dot{M} = \sqrt{\frac{\mu_\oplus}{a^3}} \left[1 - \frac{3}{2} J_2 \left(\frac{R_\oplus}{p}\right)^2 \sqrt{1-e^2} \left(\frac{3}{2}\sin^2 i - 1\right)\right] \tag{2.5}$$

where $R_\oplus = 6378.14\,\text{km}$ is the mean radius of the Earth, $p = a(1-e^2)$ is the semi-latus rectum, $\mu_\oplus = 398\,600.44\,\text{km}^3\text{s}^{-2}$ is the standard gravitational parameter of the Earth, and $J_2 = 0.00108263$ is the zonal harmonic coefficient due to the equatorial bulge of the Earth [15].

A period ratio τ is defined as a ratio of N_P/N_D and further can be deduced based on the perturbed orbital elements:

$$\tau = \frac{N_\text{P}}{N_\text{D}} = \frac{T_\text{G}}{T_\text{S}} = \frac{\dot{\omega} + \dot{M}}{\omega_\oplus - \dot{\Omega}} \tag{2.6}$$

The period ratio is used to identify a unique RGT orbit out of an N_P and N_D pair [23]. That is, a satellite orbit with $\tau = 10/2$ and a satellite orbit with $\tau = 5/1$ both of which share an identical orbit and a ground track.

The semi-major axis a of an RGT orbit can be derived using the Newton-Raphson method presented by Bruccoleri for a given set of N_P, N_D, e, and i [28]. Because the semi-major axis is a function of τ, e, and i (i.e., $a = a(\tau, e, i)$), we shall utilize the period ratio $\tau = N_\text{P}/N_\text{D}$ as an independent orbital variable instead of the semi-major axis a. Henceforth, this chapter utilizes an RGT orbital elements vector, $\mathbf{\mathfrak{oe}} = [\tau, e, i, \omega, \Omega, M]^T$, to fully define an RGT orbit of a satellite. We assume the utilization of satellite maneuvers to correct and maintain an identical ground track throughout the satellite lifetime, negating perturbation effects other than the J_2 effect. Note that the right ascension of the ascending node (RAAN) Ω and the mean anomaly M in the RGT orbital elements vector indicate the initial values

in reference to a given epoch.

2.3.2 Common Ground Track Constellation

This chapter considers a constellation pattern where all satellites in the constellation are systematically generated such that their ground tracks overlap to create a single common ground track. In this chapter, we refer to this type of constellation as a *common ground track constellation*. (If there are multiple sub-constellations, each sub-constellation has its own common ground track.) Figure 2.1 illustrates an example of arbitrarily defined 9-satellite common ground track constellation; its system satellites, depicted in yellow circles, are placed along a common ground track. The definitions of the terms used in Figure 2.1 are discussed in Section 2.4. For more information about the expanded ground track view, refer to Appendix A.1.

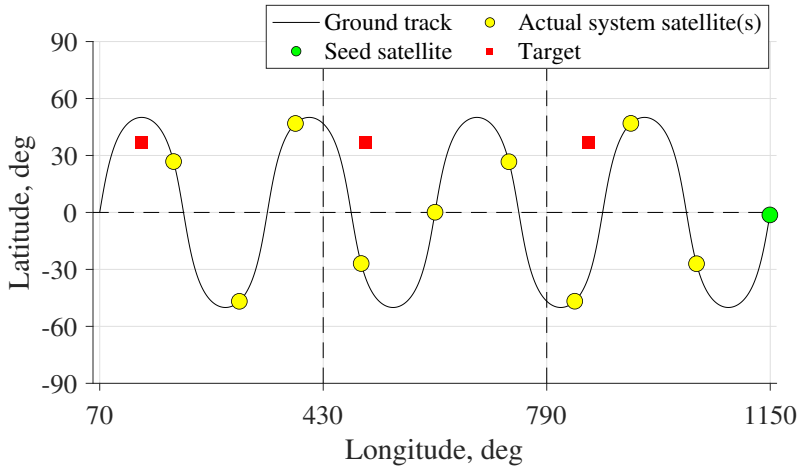


Figure 2.1: Illustration of a common ground track constellation in an expanded ground track view.

A common ground track constellation has relationships with several constellation design theories. For example, when certain conditions are satisfied (e.g., symmetric distribution and $N_D = 1$), the common ground track constellations utilizing circular RGT orbits can be expressed as $i : N/N/(N - N_P)$ when $N_D = 1$ [29], following the standard Walker notation $i : N/P/F$. Here, N is the total number of satellites in the system, P is the number

of orbital planes, and F is the Walker phasing factor.

A common ground track constellation with RGT orbits is a common assumption used in the literature such as the original Flower Constellation theory [23]. The Flower Constellation is defined as a set of N satellites following the same (closed) trajectory with respect to a rotating frame. For this chapter, the ECEF frame is considered. Ref. [30] introduces the three conditions to construct a Flower Constellation as follows:

1. The orbital period of each satellite is a rational multiple of the period of the rotating frame.

2. The semi-major axis a, eccentricity e, inclination i, and argument of perigee ω are identical for all the satellite orbits.

3. The right ascension of the ascending node Ω_k and the mean anomaly M_k of each satellite ($k = 1, ..., N$) satisfy:

$$N_{\mathrm{P}}\Omega_k + N_{\mathrm{D}}M_k = \text{constant mod } (2\pi) \tag{2.7}$$

This chapter utilizes the above three conditions of the original Flower constellation set theory as a basis for constellation generations. Furthermore, we restrict satellite orbits to be either circular or critically-inclined elliptic ($i = 63.4^\circ$ or 116.6°). This is because, in engineering practice, non-critically-inclined elliptic orbits are generally avoided for periodic coverage requirements due to heavy orbital maintenance costs incurred by negating the precession of the argument of perigee.

2.4 Circular Convolution Formulation

Building upon the satellite constellation model in the previous section, this section introduces the main ideas behind the methods developed in this chapter, including the definitions and concepts of the access profile, coverage, and constellation pattern representation,

as well as the mathematical representation of the circular convolution phenomenon.

The derivation of the circular convolution phenomenon utilizes time discretization. One underlying assumption is that, to satisfy the periodically time-varying coverage requirements, the repeat period of the RGT orbit can be chosen such that it is a rational multiple of the repeat period of the coverage requirement. This implies that we can discretize both of these repeat periods by a common time step length t_{step}. The least common multiple of the numbers of time steps for these two repeat periods would be the number of time steps for the simulation time horizon length L needed to evaluate the coverage. If there are multiple target points with different repeat periods for their coverage requirements, assuming that their repeat periods can each be represented as an integer number of time steps with the common interval t_{step}, then we can use the least common multiple of these time steps as the repeat period of the "overall" coverage requirement. The circular convolution formulation and its associated properties are defined over this discretized L-step simulation time horizon length.

For simplicity, in this chapter, we consider the case in which the repeat period of RGT is an integer multiple of the repeat period of the coverage requirement; in this case, $T_{\text{sim}} = T_{\text{r}} = Lt_{\text{step}}$, where T_{sim} is the length of the simulation time horizon and T_{r} is the repeat period of the orbit. This case can be easily generalized to the above more general case. Note that the uniformly continuous coverage case can be treated as a special case, where the repeat period of the orbit (and thus the simulation time horizon) can be arbitrarily chosen.

2.4.1 Access Profile

The relative position vector $\boldsymbol{\rho}$ pointing from a ground target point to a satellite is defined as:

$$\boldsymbol{\rho} = \boldsymbol{r}_{\text{s}} - \boldsymbol{r}_{\text{g}} \tag{2.8}$$

where $\boldsymbol{r}_s$ is a satellite position vector from the center of the Earth and $\boldsymbol{r}_g$ is a target point position vector from the center of the Earth. Figure 2.2 illustrates this relationship.

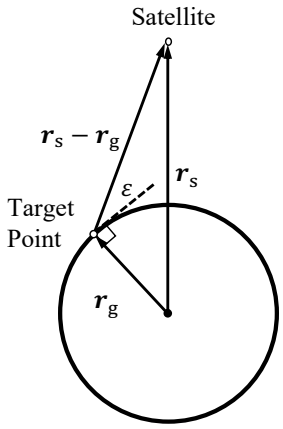


Figure 2.2: Satellite, target point, and elevation angle relationship.

An elevation angle ε of a satellite seen from a ground target point is defined as:

$$\varepsilon = \sin^{-1}\left(\frac{\boldsymbol{r}_g \cdot \boldsymbol{\rho}}{\|\boldsymbol{r}_g\|\|\boldsymbol{\rho}\|}\right) = \sin^{-1}(\hat{\boldsymbol{r}}_g \cdot \hat{\boldsymbol{\rho}}) \tag{2.9}$$

where $\|\cdot\|$ is the Euclidean norm.

Because the dot product between the unit target point position vector $\hat{\boldsymbol{r}}_g$ and the unit relative position vector $\hat{\boldsymbol{\rho}}$ continues to change due to the rotation of the Earth and the motion of a satellite, the elevation angle is, therefore, a function of time, $\varepsilon = \varepsilon(t)$. An example of a typical NGSO satellite elevation angle function is shown in the upper part of Figure 2.3. When the elevation angle of a satellite is above the minimum elevation angle threshold $\varepsilon_{\min}$, which is determined by the mission requirement [5], the satellite is said to be visible from or *to have access to the target point*. Since the periods when the satellite has access to the ground target point are of particular interest, we convert the elevation angle function into an *access profile* (or a visibility profile in some literature), which is a binary vector that indicates either access, 1, or no access, 0, at each time instant. The access profile is visualized in the lower part of Figure 2.3. This chapter utilizes a sampling method to generate an access profile. Note that access profiles can be derived in different ways [31,

32, 33].

The continuous-time elevation angle function $\varepsilon(t)$ is sampled at every time step of t_{step} to create a discrete-time elevation angle function $\varepsilon[n]$ with length L. As mentioned earlier, L is the number of time steps of the simulation horizon, i.e., $T_{\text{sim}} = Lt_{\text{step}}$, where T_{sim} is the simulation time horizon (which is assumed to be equal to the RGT repeat period T_{r} in this chapter for simplicity as discussed above). The access profile $\boldsymbol{v}_{k,j} \in \mathbb{Z}_2^L$ between the kth satellite and the jth target point stores boolean information about the satellite access (or visibility) state at each discrete-time instant $n \in \{0, ..., L-1\}$. Therefore, each element of the access profile is:

$$v_{k,j}[n] \coloneqq \begin{cases} 1, & \text{if } \varepsilon_{k,j}[n] \geq \varepsilon_{k,j,\min}[n] \\ 0, & \text{otherwise} \end{cases} \tag{2.10}$$

where n is the discrete-time instant and $\mathcal{J}$ is the set of target points. Throughout this chapter, vectors are represented in italic boldface (e.g., $\boldsymbol{v}_{k,j}$) and their elements are represented in brackets (e.g., $v_{k,j}[n]$). To make the notation consistent with the circular convolution method from the digital signal processing community, the vector index representing the discrete-time instant n is set to take the range of $[0, L-1]$.

It is important to note a condition in Eq. (2.10): there must exist at least one access interval for a given satellite-target for the methods introduced in this chapter to function; simply stated, the access profile shall be a non-zero vector. The methods introduced in the following sections are constructed based on the assumption that the access profile is a non-zero vector.

One can interpret the generalized minimum elevation angle $\varepsilon_{k,j,\min}[n]$ in Eq. (2.10) as the minimum elevation angle threshold imposed on an access between a satellite k and a target point j at discrete-time instant n. This chapter assumes that all satellites have a common generalized minimum elevation angle:

$$\varepsilon_{k,j,\min}[n] = \varepsilon_{j,\min}[n], \quad \forall k \in \{1, ..., N\} \tag{2.11}$$

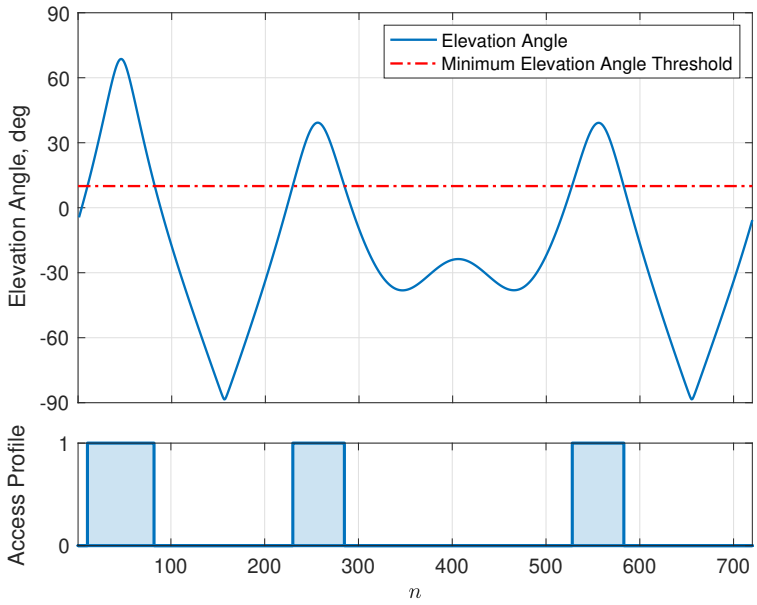


Figure 2.3: Sample illustration of a satellite's elevation angle viewed from a ground point and corresponding access profile.

When designing a satellite constellation for regional coverage, a constellation must be spatially and temporally referenced relative to the target point and the epoch. A hypothetical satellite that conveys referenced orbital information, $\mathbf{oe}_0 = [\tau, e, i, \omega, \Omega_0, M_0]^T$, for the constellation is defined as the *seed satellite*[1] and the corresponding $\mathbf{oe}_0$ as the *seed satellite orbital elements vector*. The actual satellites inherit the common orbital characteristics defined in this seed satellite elements vector, but they independently hold (Ω_k, M_k) pairs that are determined by Eq. (2.7), resulting in the orbital elements vector for each satellite of $\mathbf{oe}_k = [\tau, e, i, \omega, \Omega_k, M_k]^T$ where k is an index of a satellite (Ω and M are initial values referenced to a given epoch; the subscripts refer to the index of a corresponding satellite). Note that it is not required to have an actual satellite at the seed satellite position; the seed satellite orbital elements are used as a reference to define the actual satellites in the system.

Let us recall the main assumptions considered thus far: (1) all satellites are placed on a common repeating ground track constellation as shown in Figure 2.1; and (2) all access between a target point j and every member satellite in a given constellation are constrained

[1]The term seed satellite is credited to the software Systems Tool Kit (STK) [34].

to the same minimum elevation angle threshold as shown in Eq. (2.11). Such assumptions enable us to utilize a powerful property, a *cyclic property*, in which all access profiles of the member satellites in a given constellation are *identical, but circularly shifted*. Therefore, any access profile $\boldsymbol{v}_{k,j}$ between the kth satellite and the jth target point can be represented as a *circularly shifted seed satellite access profile* $\boldsymbol{v}_{0,j}$:

$$v_{k,j}[n] = \boldsymbol{P}_{\pi}^{n_k} v_{0,j}[n] \tag{2.12}$$

where $\boldsymbol{P}_{\pi}$ is a permutation matrix with the dimension $(L \times L)$ as shown in Eq. (2.13) and n_k is the index representing its (temporal) location of the kth satellite with respect to the seed satellite along the common ground track.

$$\boldsymbol{P}_{\pi} = \begin{bmatrix} 0 & 0 & 0 & \cdots & 1 \\ 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix} \tag{2.13}$$

The formal definition and the physical interpretation of n_k are explained in Section 2.4.3.

2.4.2 Coverage Timeline and Coverage Requirement

Because there are multiple satellites in the constellation system, the access profiles must be meshed together to create a *coverage timeline* over a target point. Hence, a coverage timeline $\boldsymbol{b}_j \in \mathbb{Z}_{\geq 0}^{L}$ is an access profile between multiple satellites and the jth target point; it stores information about the number of satellites in view at each discrete-time instant n. This is illustrated in Figure 2.4. As before, n_1 and n_2 are the indices that represent the temporal locations of the first ($k = 1$) and second ($k = 2$) satellite with respect to the seed satellite ($k = 0$), respectively. It is important to point out that because the seed satellite is hypothetical, its access profile is not considered in the coverage timeline. Eq. (2.14)

provides a mathematical definition of the coverage timeline:

$$b_j[n] = \sum_{k=1}^{N} v_{k,j}[n] \tag{2.14}$$

Note that the coverage timeline is not a binary vector, but instead, it is a non-negative integer vector.

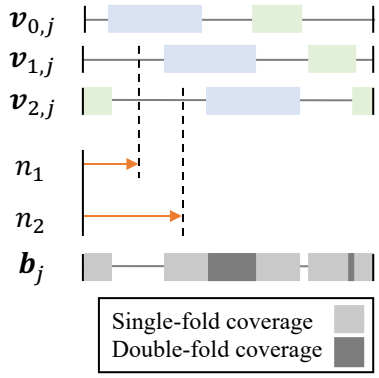


Figure 2.4: Illustration of shifts of access profiles (2-satellite system); notice that the seed satellite access profile ($\boldsymbol{v}_{0,j}$) is not part of the coverage timeline.

Next, we define the *coverage requirement*. A coverage requirement $\boldsymbol{f}_j \in \mathbb{Z}_{\geq 0}^{L}$ is a vector of non-negative integers that is created by a user per mission requirement. It is important to distinguish the difference between the coverage timeline $\boldsymbol{b}_j$ and the coverage requirement $\boldsymbol{f}_j$. The coverage timeline is a coverage performance or a state of a constellation system, whereas the coverage requirement indicates what a constellation system shall achieve. For example, in order for a constellation system to achieve f-fold continuous coverage, the coverage timeline must be greater than or equal to the coverage requirement, that is, at least f satellite(s) must have access to or be visible by the target point throughout the simulation time horizon. The coverage satisfactoriness indicator c_j indicates the coverage requirement satisfactoriness of the coverage timeline over a target point j:

$$c_j := \begin{cases} 1, & \text{if } b_j[n] \geq f_j[n], \quad \forall n \in \{0, ..., L-1\} \\ 0, & \text{otherwise} \end{cases} \tag{2.15}$$

If an area of interest consists of multiple target points (e.g., due to area grid discretization), the coverage is satisfactory if all target points are satisfactorily covered. Extending Eq. (2.15), the satisfactory condition of the coverage over all target points in a set $\mathcal{J}$ can be expressed as

$$c_{\mathcal{J}} \coloneqq \begin{cases} 1, & \text{if } c_j = 1, \ \ \forall j \in \mathcal{J} \\ 0, & \text{otherwise} \end{cases} \tag{2.16}$$

where $\mathcal{J}$ is a set of target points. Thus, designers of the constellation system must aim to satisfy all coverage requirements on every target point as each target point may impose its own unique coverage requirement.

2.4.3 Constellation Pattern Vector

We express the time shifts of satellites along the ground track with respect to the seed satellite in a discrete-time binary sequence $\boldsymbol{x} \in \mathbb{Z}_2^L$ and refer to it as the *constellation pattern vector*:

$$x[n] \coloneqq \begin{cases} 1, & \text{if } n = n_k \\ 0, & \text{otherwise} \end{cases} \tag{2.17}$$

The temporal location index, n_k, can be interpreted as the time-delay index for the kth satellite. This is because the kth satellite that is delayed behind the seed satellite by the time difference of $\Delta t_k = t_{\text{step}} n_k$ over the common ground track can be equivalently shown as a unit impulse at time instant $n = n_k$ on a constellation pattern vector $\boldsymbol{x}$. This is illustrated in Figure 2.5. The left-hand side of the figure shows a snapshot of an arbitrary constellation system: a seed satellite depicted as the green circle and an arbitrary kth satellite depicted as the yellow circle in an expanded ground track view. The kth satellite is positioned behind the seed satellite in a moving direction by the time unit of Δt_k. That is, the kth satellite will occupy the current position of the seed satellite Δt_k time units later (i.e., n_k time steps later). The equivalent representation in the constellation pattern vector form is shown on

the right-hand side of the figure. In this case, the position of the kth satellite is represented as a red impulse, which represents the time delay with respect to the seed satellite.

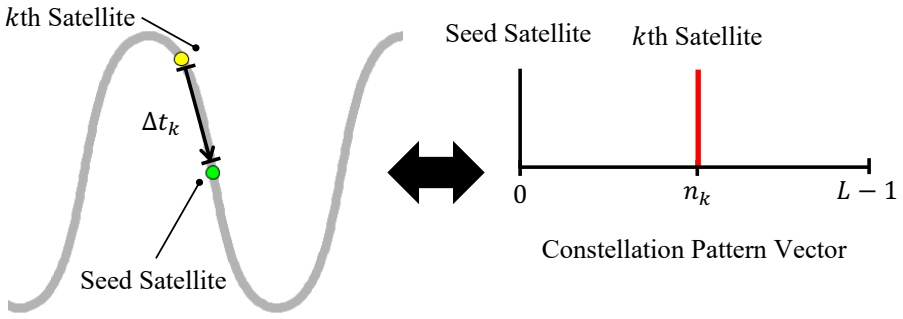


Figure 2.5: Illustration of a satellite time shift and its representation in the constellation pattern vector form; the direction of the motion of satellites is indicated by the arrow on the left-hand side of the figure.

From Eq. (2.17) and because L is assumed to cover exactly one repeat period of the RGT, we can deduce the total number of satellites in the constellation from the constellation pattern vector as

$$N = \sum_{n=0}^{L-1} x[n] \tag{2.18}$$

2.4.4 Circular Convolution Phenomenon

The discrete-time sequences, $\boldsymbol{v}_{0,j}$, $\boldsymbol{x}$, and $\boldsymbol{b}_j$, defined in the previous sections have a finite periodic length of L due to the cyclic property of the closed relative ground track assumption. Note that, as mentioned earlier, this length of the vectors is the total number of time steps for the simulation time horizon.

A discrete circular convolution operation between the seed satellite access profile $\boldsymbol{v}_{0,j}$ and the constellation pattern vector $\boldsymbol{x}$ produces a coverage timeline $\boldsymbol{b}_j$:

$$\begin{aligned} b_j[n] = v_{0,j}[n] \circledast x[n] &= \sum_{m=0}^{L-1} v_{0,j}[m]x[(n-m) \bmod L] \\ = x[n] \circledast v_{0,j}[n] &= \sum_{m=0}^{L-1} x[m]v_{0,j}[(n-m) \bmod L] \end{aligned} \tag{2.19}$$

where $\circledast$ represents a circular convolution operator. (Note that the circular convolution is commutative.) Or equivalently, this equation can be written as

$$\boldsymbol{V}_{0,j}\boldsymbol{x} = \boldsymbol{b}_j \tag{2.20}$$

where $\boldsymbol{V}_{0,j} \in \mathbb{Z}_2^{L\times L}$ is a seed satellite access profile circulant matrix that is fully specified by a seed satellite access profile $\boldsymbol{v}_{0,j}$. Note that a circulant matrix is a special form of a Toeplitz matrix [35]; each entry of the matrix $[\alpha,\beta]$ is defined as

$$V_{0,j}[\alpha,\beta] = v_{0,j}[(\alpha-\beta) \bmod L] \tag{2.21}$$

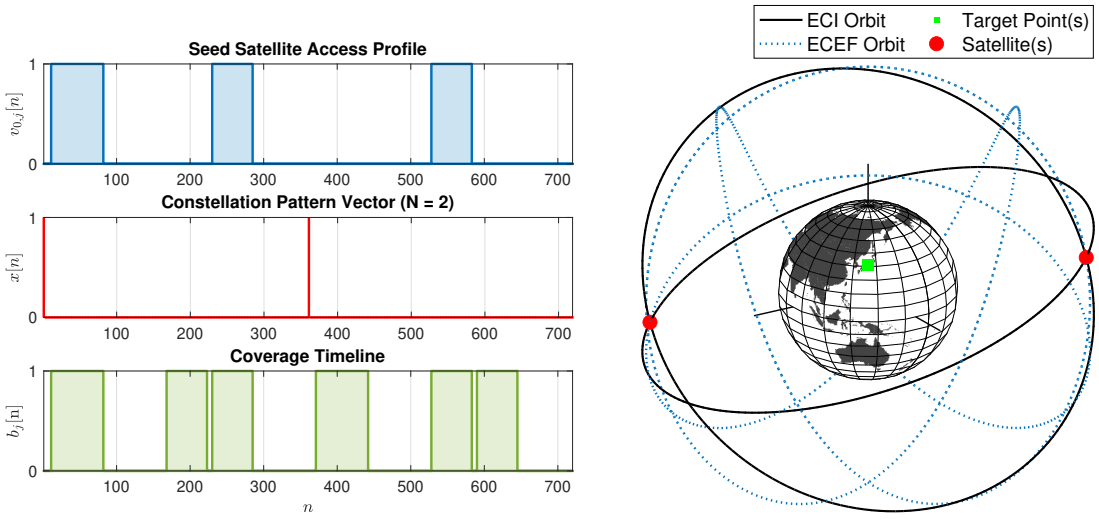
where α and β are the row and column indices, respectively, for $\alpha,\beta \in \{0,1,\cdots,L-1\}$. More information about the circular convolution is referred to Ref. [36]. The derivation of the circular convolution relationship [Eq. (2.20)] from Eq. (2.14) is described in Appendix A.2.

To illustrate this relationship, consider a system with $\mathbf{\text{œ}}_0 = [4/1, 0, 50^\circ, 0^\circ, 350.2^\circ, 0^\circ]^T$ (J2000) and uniformly spaced $N = 2$ satellites. The corresponding seed satellite access profile observed from a target $\mathcal{J} = \{(\phi = 36.7^\circ\text{N}, \lambda = 137.48^\circ\text{E})\}$ (a point) with $\varepsilon_{\min} = 10^\circ$ is shown in the top part of Figure 2.6a. In this example, the length of vectors is set to $L = 720$ such that the corresponding time step is approximately 120 s. The constellation pattern vector, shown in the middle part of Figure 2.6a, has two unit impulses at $n = 0$ and $n = 360$ to represent the temporal locations of two satellites with respect to the seed satellite. The equivalent orbital elements vectors for these satellites are (refer to Section 2.5.4 for the derivation):

$$\mathbf{\text{œ}}_1 = [4/1, 0, 50^\circ, 0^\circ, 350.2^\circ, 0^\circ]^T$$
$$\mathbf{\text{œ}}_2 = [4/1, 0, 50^\circ, 0^\circ, 170.2^\circ, 0^\circ]^T$$

In this case, the first satellite of the system is essentially identical to the seed satellite

(i.e., a unit impulse at $n = 0$). The circular convolution between the seed satellite access profile and the constellation pattern vector yields the coverage timeline shown in the bottom part of Figure 2.6a. A snapshot of the corresponding configuration in the Earth-centered inertial (ECI) and Earth-centered Earth-fixed (ECEF) frame at $n = 0$ is shown in Figure 2.6b.



(a) The APC decomposition.

(b) The corresponding configuration in the ECI and ECEF frame at $n = 0$.

Figure 2.6: APC decomposition and its equivalent constellation representation in 3D space.

This formulation exhibits the satellite constellation architecture by laying out the relationships between the common orbital characteristics, the satellite constellation pattern, and the coverage performance. We shall hereafter refer to this type of satellite constellation design decomposition into three vectors $\boldsymbol{v}_{0,j}$, $\boldsymbol{x}$, and $\boldsymbol{b}_j$ as the *APC decomposition*, following the acronyms of the seed satellite Access profile, constellation Pattern, and Coverage timeline. Methods that are derived based on the APC decomposition are called the *APC-based methods*.

2.5 Regional Coverage Constellation Pattern Design Methods

2.5.1 Problem Statement

Following the APC decomposition introduced herein, the satellite constellation design can be split into defining the reference seed satellite orbital elements $\mathbf{oe}_0$ (which includes the common orbital characteristics) and defining the constellation pattern vector $\boldsymbol{x}$. Conventional methods often make simple assumptions for $\boldsymbol{x}$ such as a symmetric pattern (e.g., Walker constellations) and optimize $\mathbf{oe}_0$; instead, this chapter focuses on the optimization of the $\boldsymbol{x}$ itself without such simplifying assumptions. Mathematically, the goal of this chapter is to solve for the optimal constellation pattern vector $\boldsymbol{x}^*$ such that the coverage timeline $\boldsymbol{b}_j^* = \boldsymbol{v}_{0,j} \circledast \boldsymbol{x}^*$ is equal to or greater than the designated $\boldsymbol{f}$ coverage threshold. The objective function is the number of satellites required N, which can be deduced from Eq. (2.18). The seed satellite orbital elements $\mathbf{oe}_0$ is considered as a given input so that the developed constellation pattern design approach can be integrated with the existing established methods for determining $\mathbf{oe}_0$ (e.g., brute-force methods, genetic algorithms). Appendix A.3 introduces an example approach to integrate the determination of the seed satellite orbital elements $\mathbf{oe}_0$ and the design of the satellite constellation pattern design $\boldsymbol{x}$.

This section introduces two constellation pattern optimization methods based on the circular convolution formulation and APC decomposition. First, we derive a rather conventional iterative method using a common assumption of symmetry; this method is used as a baseline for later analysis. Next, we develop a novel and general method based on binary integer linear programming to perform rigorous optimization of the constellation pattern.

2.5.2 Baseline: Quasi-Symmetric Method

The baseline quasi-symmetric method aims to design the satellite constellation pattern with uniform temporal spacing between satellites along the common closed trajectory in space.

Given a length L of the constellation pattern vector, the uniform temporal spacing constant $\eta \in \mathbb{R}_{>0}$ between satellites is defined as

$$\eta \coloneqq \frac{L}{N} \tag{2.22}$$

We first consider a special case where η is an integer. In this case and assuming $n_1 = 0$, we can construct a symmetric constellation pattern (i.e., a uniform distribution of satellites along the common ground track of the constellation system) using the following the constellation pattern vector form:

$$\bar{x}[n] \coloneqq \sum_{k=1}^{N} \delta[n - \eta(k-1)] \tag{2.23}$$

where

$$\delta[n] = \begin{cases} 1, & \text{if } n = 0 \\ 0, & \text{otherwise} \end{cases} \tag{2.24}$$

A user is allowed to arbitrarily set the temporal location of the first satellite n_1 ($0 \leq n_1 < L$). In this case, Eq. (2.23) requires a circular shift of $\bar{x}[n]$:

$$x[n] = \bar{x}[n] \circledast \delta[n - n_1] \tag{2.25}$$

Next, we generalize this formulation into the case where η is not an integer. In this case, we cannot achieve a strictly symmetric constellation pattern with the given discretization, but only a near-symmetric one; we call the latter a *quasi-symmetric* constellation pattern in this chapter. For this generalization, the only change we need to make is to replace Eq. (2.23) by Eq. (2.26):

$$\bar{x}[n] \coloneqq \sum_{k=1}^{N} \delta[\text{nint}(n - \eta(k-1))] \tag{2.26}$$

where nint(·) is the nearest integer function, which is used to guarantee the integer-indexing of a vector.

Algorithm 1 is designed to perform an iterative search about N and n_1 until the coverage requirement is satisfied and outputs the optimal constellation pattern vector $\boldsymbol{x}^*$ given a set of $\boldsymbol{v}_{0,j}$ and $\boldsymbol{f}_j$. The algorithm consists of two nested iterative loops. The outer loop increments N by one at each iteration, whereas the inner loop performs an exhaustive search about n_1 to find the N-minimizing temporal location of the first satellite. (Note that the range for the inner loop is set to $0 \leq n_1 \leq \text{nint}(\eta) - 1$ due to the (quasi-)symmetry of the resulting constellation pattern vector.) These loops break when the coverage requirement is satisfied as shown in Algorithm 1. If no quasi-symmetric constellation is found until the outer loop for N reaches the maximum number of satellites, which is equal to L, the method would determine the problem to be infeasible.

Algorithm 1: Quasi-symmetric method to compute $\boldsymbol{x}^*$, $\boldsymbol{b}_j^*$, N, and n_1 (point-coverage)

Input: $\boldsymbol{v}_{0,j}$, $\boldsymbol{f}_j$
Output: $\boldsymbol{x}^*$, $\boldsymbol{b}_j^*$, N, and n_1

```
1  N = 1
2  while True do
3      if N ≤ L then
4          Generate x̄[n] based on η := L/N as outlined in Eq. (2.23)
5          for n_1 = 0, ..., nint(η) − 1 do
6              Generate x[n] based on x̄[n] and n_1 as outlined in Eq. (2.25)
7              Compute b_j[n] = v_{0,j}[n] ⊛ x[n] via Eq. (2.19)
8              if c_j = 1 as in Eq. (2.15) then
9                  Break the loops
10                 return x*, b_j*, N [Eq. (2.18)], and n_1
11         N = N + 1
12     else
13         return Infeasible
```

For an area of interest consisting of multiple target points, a user may replace line 9 in Algorithm 1 with "**if** $c_{\mathcal{J}} = 1$ as in Eq. (2.16) **then**". This guarantees the iterative search until all target points are satisfactorily covered. Similarly, one can come up with a custom

termination criterion and/or figure of merit, such as time percent coverage or area percent coverage metrics. This is feasible since each iteration provides a full coverage state across all target points.

An overview of the quasi-symmetric method is shown in Figure 2.7. The seed satellite orbital elements vector, minimum elevation angle, reference epoch, and a set of target points are the user-defined parameters, which are determined based on mission requirements.

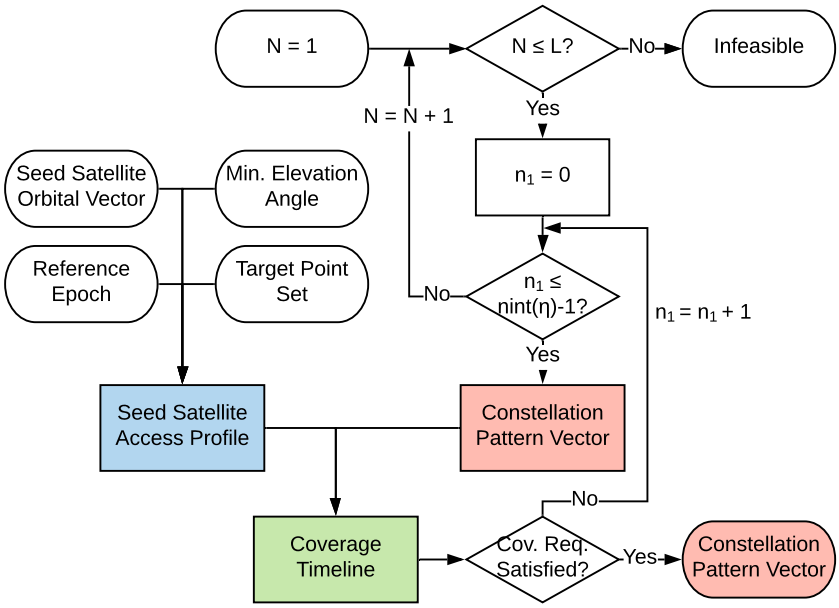


Figure 2.7: Overview of the quasi-symmetric method.

2.5.3 New Method: Binary Integer Linear Programming (BILP) Method

This subsection introduces the new satellite constellation pattern method developed in this chapter using BILP. The BILP method aims to optimize the constellation pattern in a more rigorous and general way, without assuming symmetry and, if needed, concurrently considering multiple sub-constellations. Recall Eq. (2.20):

$$\boldsymbol{V}_{0,j}\boldsymbol{x} = \boldsymbol{b}_j \tag{2.27}$$

where $\boldsymbol{V}_{0,j} \in \mathbb{Z}_2^{L\times L}$ is a circulant matrix that is fully specified by the seed satellite access profile $\boldsymbol{v}_{0,j}$ as shown in Eq. (2.21). This definition of $\boldsymbol{V}_{0,j}$ can be expanded as Eq. (2.28):

$$\boldsymbol{V}_{0,j} = \begin{bmatrix} v_{0,j}[0] & v_{0,j}[L-1] & v_{0,j}[L-2] & \cdots & v_{0,j}[1] \\ v_{0,j}[1] & v_{0,j}[0] & v_{0,j}[L-1] & \cdots & v_{0,j}[2] \\ v_{0,j}[2] & v_{0,j}[1] & v_{0,j}[0] & & \vdots \\ \vdots & \vdots & \ddots & \ddots & \\ v_{0,j}[L-1] & v_{0,j}[L-2] & \cdots & & v_{0,j}[0] \end{bmatrix} \tag{2.28}$$

Each column of a circulant matrix $\boldsymbol{V}_{0,j}$ is identical to a circularly-shifted seed satellite access profile $\boldsymbol{v}_{0,j}$. Eq. (2.27) can be shown in a matrix form:

$$\begin{bmatrix} v_{0,j}[0] & v_{0,j}[L-1] & v_{0,j}[L-2] & \cdots & v_{0,j}[1] \\ v_{0,j}[1] & v_{0,j}[0] & v_{0,j}[L-1] & \cdots & v_{0,j}[2] \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ v_{0,j}[L-1] & v_{0,j}[L-2] & \cdots & & v_{0,j}[0] \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ \vdots \\ x[L-1] \end{bmatrix} = \begin{bmatrix} b_j[0] \\ b_j[1] \\ \vdots \\ b_j[L-1] \end{bmatrix} \tag{2.29}$$

An interesting observation can be formalized. If we are given $\boldsymbol{v}_{0,j}$ and $\boldsymbol{x}$, then we can produce $\boldsymbol{b}_j$—this is the assumption of the quasi-symmetric method at each iteration. Likewise, if $\boldsymbol{v}_{0,j}$ and $\boldsymbol{b}_j$ are given, then we can analytically solve for $\boldsymbol{x}$ by solving the system of linear equations in Eq. (2.27) and obtain $\boldsymbol{x} = \boldsymbol{V}_{0,j}^{-1}\boldsymbol{b}_j$ ($\det(\boldsymbol{V}_{0,j}) \neq 0$). Since $\boldsymbol{b}_j$ represents the entire coverage timeline, this analysis enables us to find a constellation pattern vector $\boldsymbol{x}$ that satisfies a given coverage requirement $\boldsymbol{f}_j$.

Although this approach provides us with a way to find the satellite constellation pattern, the resulting $\boldsymbol{x}$ is not necessarily a binary vector, which violates the nature of the constellation pattern vector. The existence of a satellite at a given instance cannot be represented in a decimal number but only as either one or zero. Therefore, to guarantee a physical quantification of satellites, we shall employ the binary integer linear programming, or BILP, to

solve for $\boldsymbol{x}^*$ which satisfies the inequality constraint:

$$\boldsymbol{V}_{0,j}\boldsymbol{x}^* = \boldsymbol{b}_j^* \geq \boldsymbol{f}_j \tag{2.30}$$

Before we formalize the BILP problem that solves Eq. (2.30), we introduce linear properties associated with Eq. (2.27).

Multiple Target Points

Because the system is linear, we can extend Eq. (2.27) to an area of interest that consists of *multiple target points*:

$$\begin{bmatrix} \boldsymbol{V}_{0,1} \\ \boldsymbol{V}_{0,2} \\ \vdots \\ \boldsymbol{V}_{0,|\mathcal{J}|} \end{bmatrix} \boldsymbol{x} = \begin{bmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \\ \vdots \\ \boldsymbol{b}_{|\mathcal{J}|} \end{bmatrix} \tag{2.31}$$

where $|\mathcal{J}|$ is the cardinality of a target point set $\mathcal{J}$. Eq. (2.31) has the dimension of $(|\mathcal{J}|L \times L) \cdot (L \times 1) = (|\mathcal{J}|L \times 1)$.

The augmented circulant matrix on the left-hand side is a matrix of matrices obtained by appending all circulant matrices $\boldsymbol{V}_{0,1}, ..., \boldsymbol{V}_{0,|\mathcal{J}|}$ linearly. Similarly, the augmented coverage timeline vector is also obtained by appending all coverage timeline vectors $\boldsymbol{b}_1, ..., \boldsymbol{b}_{|\mathcal{J}|}$ linearly. Here, the constellation pattern vector $\boldsymbol{x}$ represents a single constellation configuration that satisfies the augmented linear condition.

Multiple Sub-Constellations

Another direction of linearity regarding having multiple sub-constellations is observed. We consider a constellation system consisting of *multiple sub-constellations* with different seed satellite access profiles, $\boldsymbol{v}_{0,j}^{(1)}, ..., \boldsymbol{v}_{0,j}^{(z)}, ..., \boldsymbol{v}_{0,j}^{(|\mathcal{Z}|)}$, where superscript z in parenthesis denotes the index of a sub-constellation, $\mathcal{Z}$ is a set of sub-constellations, and $|\mathcal{Z}|$ represents its cardinality. Each sub-constellation seed satellite access profile $\boldsymbol{v}_{0,j}^{(z)}$ is computed based on its seed

satellite orbital elements vector $\mathbf{oe}_0^{(z)}$ and the modified minimum elevation angle threshold $\varepsilon_{j,\min}^{(z)}[n],\ j \in \mathcal{J},\ z \in \mathcal{Z},\ n \in \{0, ..., L-1\}$, which is only applicable to the BILP method (since the quasi-symmetric method does not define multiple sub-constellations). The goal of the multiple sub-constellation system is to satisfy a common coverage requirement over a single target point j. Thus, this can be incorporated by replacing Eq. (2.27) by the following equation:

$$\begin{bmatrix} \boldsymbol{V}_{0,j}^{(1)} & \boldsymbol{V}_{0,j}^{(2)} & \cdots & \boldsymbol{V}_{0,j}^{(|\mathcal{Z}|)} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}^{(1)} \\ \boldsymbol{x}^{(2)} \\ \vdots \\ \boldsymbol{x}^{(|\mathcal{Z}|)} \end{bmatrix} = \boldsymbol{b}_j \tag{2.32}$$

where the dimension of the system is $(L \times |\mathcal{Z}|\, L) \cdot (|\mathcal{Z}|\, L \times 1) = (L \times 1)$.

To guarantee the validity of this approach, we assume a synchronization condition among the sub-constellations to guarantee synchronized repeatability of the resulting coverage timeline:

$$T_{\mathrm{r}}^{(1)} = ... = T_{\mathrm{r}}^{(z)} = ... = T_{\mathrm{r}}^{(|\mathcal{Z}|)} \tag{2.33}$$

where $T_{\mathrm{r}}^{(z)},\ z \in \mathcal{Z}$ is the period of repetition, which can be written as a function of a, e, and i and is therefore unique to each sub-constellation. Note that this does not mean that the individual orbital elements for each sub-constellation need to be all identical; instead, it only means that the period of repetition, defined by Eq. (2.1), needs to be identical.

A System of Multiple Sub-Constellations for Multiple Target Points

Combining both directions of linearity—multiple target points and multiple sub-constellations—we get the following generalized governing relationship:

$$\begin{bmatrix} \boldsymbol{V}_{0,1}^{(1)} & \boldsymbol{V}_{0,1}^{(2)} & \cdots & \boldsymbol{V}_{0,1}^{(|\mathcal{Z}|)} \\ \boldsymbol{V}_{0,2}^{(1)} & \boldsymbol{V}_{0,2}^{(2)} & \cdots & \boldsymbol{V}_{0,2}^{(|\mathcal{Z}|)} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{V}_{0,|\mathcal{J}|}^{(1)} & \boldsymbol{V}_{0,|\mathcal{J}|}^{(2)} & \cdots & \boldsymbol{V}_{0,|\mathcal{J}|}^{(|\mathcal{Z}|)} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}^{(1)} \\ \boldsymbol{x}^{(2)} \\ \vdots \\ \boldsymbol{x}^{(|\mathcal{Z}|)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \\ \vdots \\ \boldsymbol{b}_{|\mathcal{J}|} \end{bmatrix} \tag{2.34}$$

where the dimension of the system is $(|\mathcal{J}|\,L \times |\mathcal{Z}|\,L) \cdot (|\mathcal{Z}|\,L \times 1) = (|\mathcal{J}|\,L \times 1)$.

Eq. (2.34) can be expressed in an indexed equation form:

$$\sum_{z=1}^{|\mathcal{Z}|} \boldsymbol{V}_{0,j}^{(z)} \boldsymbol{x}^{(z)} = \boldsymbol{b}_j, \quad \forall\, j \in \mathcal{J} \tag{2.35}$$

where the subscript j is the target point index and the superscript z is the sub-constellation index.

The physical interpretation of Eq. (2.34) is as follows: it represents a linear relationship between the physical configuration of a system of multiple sub-constellations and the resulting coverage timelines over a set of multiple target points. Here, each sub-constellation may exhibit its own unique orbital characteristics. For example, a sub-constellation ($z = 1$) may be placed on a critically-inclined elliptic orbit while a sub-constellation ($z = 2$) may be placed on a circular low Earth orbit. Similarly, each target point may impose an independent coverage requirement. For example, a target point ($j = 1$) may require continuous single-fold coverage whereas a target point ($j = 2$) may require a sinusoidal-like time-varying coverage, fluctuating between the double and triple folds. Revisiting the inequality constraint as shown in Eq. (2.30), it is the goal of the binary integer linear programming to determine the satellite constellation configurations $\boldsymbol{x}^{(1)}, ..., \boldsymbol{x}^{(z)}, ..., \boldsymbol{x}^{(|\mathcal{Z}|)}$ that satisfy this complex relationship.

Binary Integer Linear Programming (BILP) Problem Formulation

Let us assume that we want to achieve a $\boldsymbol{f}_j$-fold coverage system ($\forall j \in \mathcal{J}$) with the given $\boldsymbol{v}_{0,j}^{(1)}, ..., \boldsymbol{v}_{0,j}^{(z)}, ..., \boldsymbol{v}_{0,j}^{(|\mathcal{Z}|)}$ vectors. The BILP formulation is shown in Eq. (2.36). Solving the most general form of the problem, Eq. (2.35), via BILP yields an optimal solution in the form of "*a system of multiple sub-constellations that simultaneously satisfies the coverage requirements over multiple target points*":

$$\begin{aligned} \underset{\boldsymbol{x}}{\text{minimize}} \quad & \sum_{z=1}^{|\mathcal{Z}|} \mathbf{1}^T \boldsymbol{x}^{(z)} \\ \text{subject to} \quad & \sum_{z=1}^{|\mathcal{Z}|} \boldsymbol{V}_{0,j}^{(z)} \boldsymbol{x}^{(z)} \geq \boldsymbol{f}_j, \quad \forall j \in \mathcal{J} \\ & \boldsymbol{x}^{(z)} \in \mathbb{Z}_2^L, \quad \forall z \in \mathcal{Z} \end{aligned} \tag{2.36}$$

where the binary design variable constraint is imposed on the elements of the constellation pattern vector $\boldsymbol{x}$ to reflect the physical quantification of satellites. The solution to this BILP problem is the optimal constellation pattern vector $\boldsymbol{x}^*$.

An overview of the BILP method is shown in Figure 2.8.

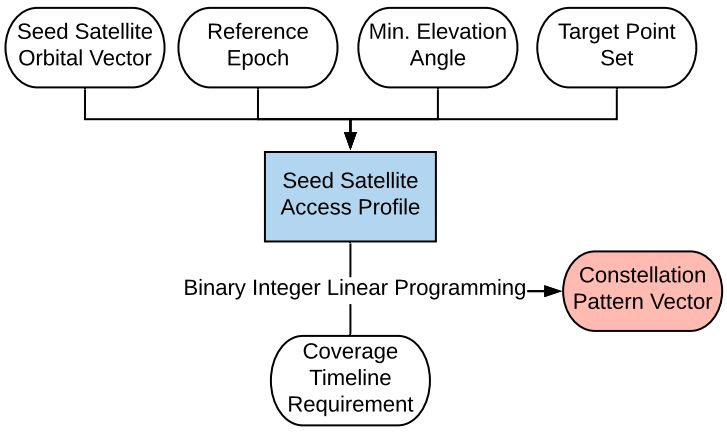


Figure 2.8: Overview of the binary integer linear programming method.

2.5.4 Derivation of Ω and M from the Constellation Pattern Vector

Once the aforementioned methods obtain an optimal constellation pattern vector, it must be post-processed to extract interpretable orbital information—a set of $(\Omega, M)_k$ where k is the index of a satellite. Every impulse on a constellation pattern vector corresponds to a point in the (Ω, M)-space. Given n_k found from the constellation pattern vector, one can find $(\Omega, M)_k$ set by solving the following system of equations:

$$N_{\mathrm{P}}(\Omega_k - \Omega_0) + N_{\mathrm{D}}(M_k - M_0) = 0 \bmod (2\pi) \tag{2.37a}$$

$$\Omega_k = n_k \frac{2\pi N_{\mathrm{D}}}{L} + \Omega_0 \tag{2.37b}$$

Note that Eq. (2.37a) is rearranged from Eq. (2.7) [30]. The derivation of Eq. (2.37b) is explained in Appendix A.4.

2.6 Illustrative Examples

This section aims to demonstrate the general applicability of and the computational efficiency associated with the proposed methods under various mission profiles. Five illustrative examples are uniquely set up by varying orbital characteristics, area of interest properties, minimum elevation angle, and coverage requirements to illustrate the APC decomposition.

All illustrative examples are conducted on an Intel Core i9-9940X Processor @3.30 GHz platform. For BILP problems, Gurobi 9.0.0 is used with the default termination setting [37]. The referenced ellipsoid model adopts the World Geodetic System 1984 (WGS 84). It is assumed that all satellites point to their nadir directions. Furthermore, we assume the utilization of satellite maneuvers to correct and maintain an identical ground track throughout the satellite lifetime, negating the perturbation effects other than the J_2 effect. Lastly,

we make an assumption that the minimum elevation angle threshold is time-invariant:

$$\varepsilon_{j,\min}[n] = \varepsilon_{j,\min}, \quad \forall n \in \{0, ..., L-1\}$$

Table 2.1 is a list of parameters used for each example study. The five examples are chosen to test different capabilities of the methods: Example 1 for single-fold continuous coverage over a single target point; Example 2 for time-varying coverage over a single target point; Example 3 for single-fold continuous coverage over multiple target points; Example 4 for time-varying and spatially-varying coverage over multiple target points; and Example 5 for multiple sub-constellations over multiple target points. All examples uniquely illustrate a variety of orbit (circular vs. critically-inclined elliptic, prograde vs. retrograde, and low vs. high altitudes) and a variety of areas of interest (a single target point vs. multiple target points and contiguous vs. discontiguous). Both the baseline quasi-symmetric method and the BILP method are applied to all examples, with an exception of the quasi-symmetric method for Example 5 due to its incapability of handling multiple sub-constellations. In this section, the subscripts *qs* and *bilp* denote variables associated with the quasi-symmetric and the BILP methods, respectively. The rest of this section discusses the details of each illustrative case.

Table 2.1: Example parameters.

Example	Seed Satellite Orbital Elements [a,b]	Min. Elev. Angle	Target Point Set	Cov. Req.	L
1	$[12/1, 0, 102.9°, 0°, 98.3°, 0°]^T$	5°	$\{(\phi = 34.75°\text{N}, \lambda = 84.39°\text{W})\}$	$\mathbf{1}$	720
2	$[12/1, 0, 102.9°, 0°, 98.3°, 0°]^T$	5°	$\{(\phi = 34.75°\text{N}, \lambda = 84.39°\text{W})\}$	Time-Varying	720
3	$[5/1, 0.41, 63.435°, 90°, 0°, 0°]^T$	30°	{Antarctica}	$\mathbf{1}$	718
4	$[83/6, 0, 99.2°, 0°, 0°, 0°]^T$	20°	$\mathcal{J}_1$ = {Amazon River Basin}, $\mathcal{J}_2$ = {Nile River Basin}	Time-Varying & Spatially-Varying	4200
5	$\mathbf{œ}_0^{(1)} = [8/1, 0, 70°, 0°, 0°, 0°]^T$ $\mathbf{œ}_0^{(2)} = [6/1, 0, 47.915°, 0°, 0°, 0°]^T$	$\varepsilon_{1,\min} = 15°$ $\varepsilon_{2,\min} = 10°$	$\{(\phi = 64.14°\text{N}, \lambda = 21.94°\text{W})$, $(\phi = 19.07°\text{N}, \lambda = 72.87°\text{E})\}$	$\mathbf{1}$	717

[a] The seed satellite orbital elements vector $\mathbf{œ}_0$ takes the form of $[\tau, e, i, \omega, \Omega_0, M_0]^T$.
[b] All orbital elements are in J2000.

2.6.1 Example 1. Single-Fold Continuous Coverage over a Single Target Point

A target point is located at $\{(\phi = 34.75°\text{N}, \lambda = 84.39°\text{W})\}$ and requires $\varepsilon_{\min} = 5°$. A seed satellite orbital elements vector $\mathbf{oe}_0 = [12/1, 0, 102.9°, 0°, 98.3°, 0°]^T$ is assumed. The period of repetition is 86 400 s. The length of vectors is selected, $L = 720$, such that the time step is 120 s. The objective is to find the optimal constellation pattern vector $\boldsymbol{x}^*$ that satisfies a single-fold continuous coverage requirement ($\boldsymbol{f} = \mathbf{1}$).

The results are obtained as follows:

$$x^*_{\text{qs}}[n] = \begin{cases} 1, & \text{for } n = \begin{matrix} 0, 33, 65, 98, 131, 164, 196, 229, 262, 295, 327, 360, 393, 425, \ldots \\ 458, 491, 524, 556, 589, 622, 655, 687 \end{matrix} \\ 0, & \text{otherwise} \end{cases}$$

$$x^*_{\text{bilp}}[n] = \begin{cases} 1, & \text{for } n = \begin{matrix} 39, 73, 79, 89, 170, 184, 234, 250, 331, 341, 347, 492, 502, \ldots \\ 542, 638, 648, 654, 663 \end{matrix} \\ 0, & \text{otherwise} \end{cases}$$

where the total number of satellites obtained for each method is $N_{\text{qs}} = 22$ and $N_{\text{bilp}} = 18$, with the computational time 0.1 s for the quasi-symmetric method and 5937.6 s for the BILP method. The results indicate that, although the computational cost for the BILP method is longer, it can explore a substantially larger design space and achieve a fewer-satellite configuration than the quasi-symmetric method by breaking the symmetry.

Figure 2.9 illustrates the (Ω, M)-space and where each of the quasi-symmetric and BILP solution constellations lies. In this example, $L = 720$; therefore, there are $L = 720$ number of *admissible* points in the (Ω, M)-space into which a satellite can be placed. Analyzing the patterns in Figure 2.9, the quasi-symmetric set depicts a lattice-like symmetry in the (Ω, M)-space whereas the BILP set exhibits asymmetry in the (Ω, M)-space.

The APC decomposition figures are shown in Figure 2.10. One can observe that the single-fold continuous coverage requirement is satisfied everywhere. Again, the asymmetry

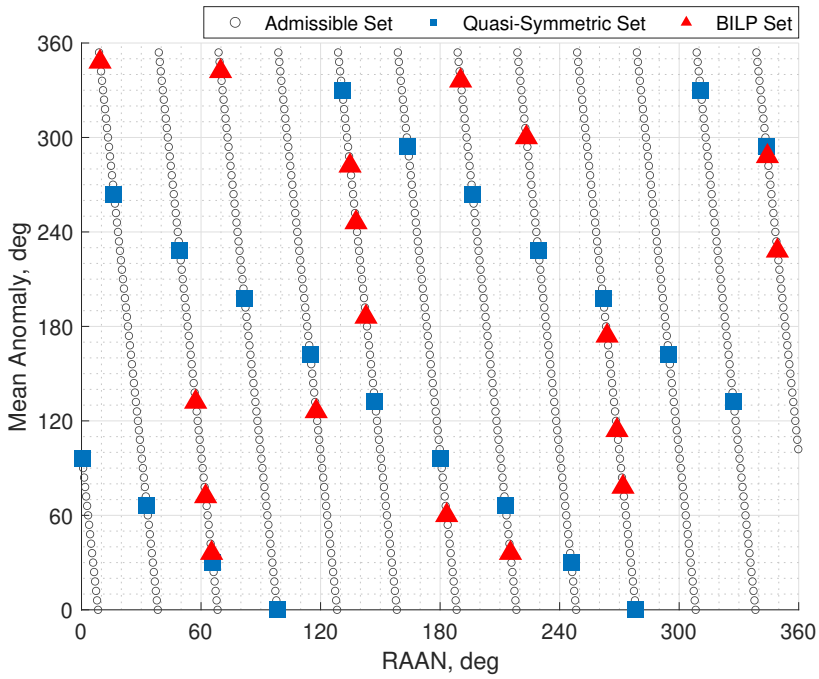


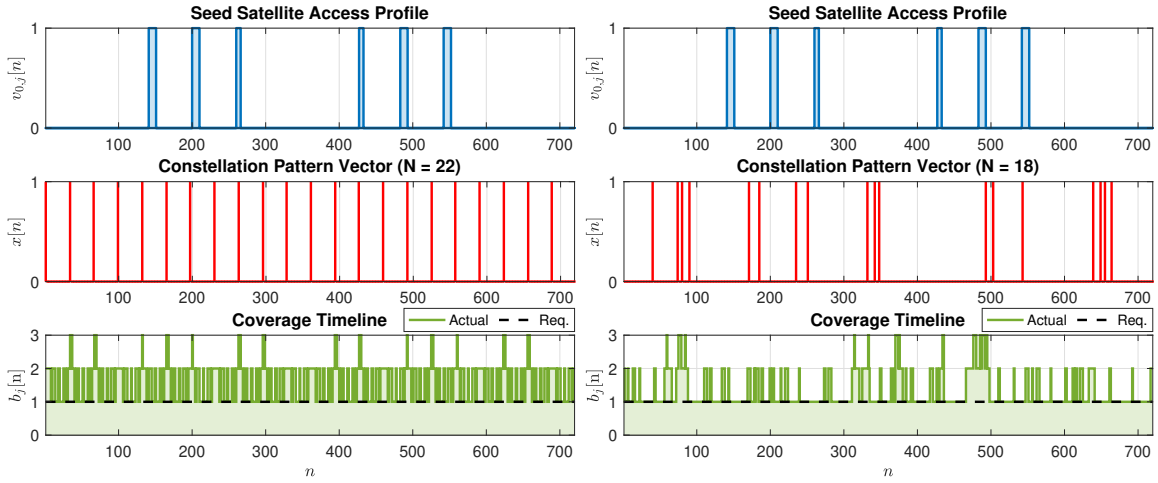
Figure 2.9: Example 1: Admissible set, quasi-symmetric set, and binary integer linear programming set in the (Ω, M)-space.

in the constellation pattern vector from the BILP method is contrasted with the symmetry in that from the quasi-symmetric method. Note that the coverage timeline for the quasi-symmetric constellation may not be strictly symmetric as η is not an integer in this case.

A snapshot of the corresponding constellation configurations at $n = 0$ is shown in Figure 2.11. This figure visually shows that the BILP method is taking advantage of the asymmetry to achieve a smaller number of satellites.

2.6.2 Example 2. Time-Varying Coverage over a Single Target Point

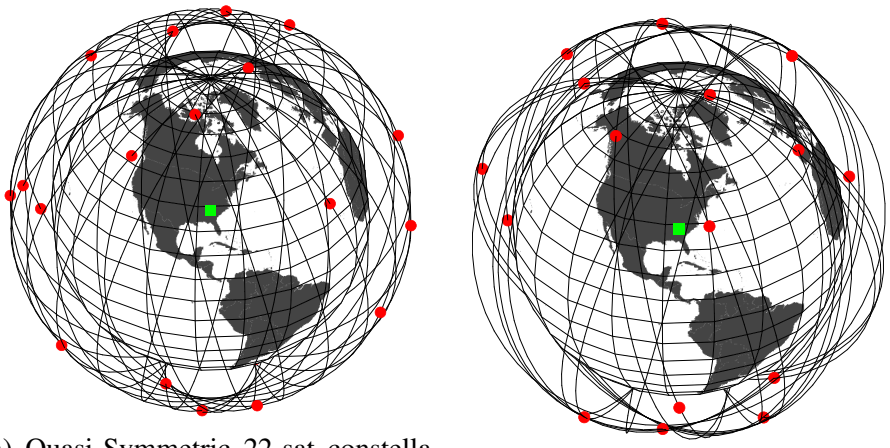
In this example, we execute a single variation to Example 1 such that the coverage requirement is now periodically time-varying with the rest of the parameters being identical (e.g., $L = 720$). The objective is to find the optimal constellation pattern vector $\boldsymbol{x}^*$ that satisfies



(a) Quasi-Symmetric method.

(b) BILP method.

Figure 2.10: Example 1: the APC decomposition.



(a) Quasi-Symmetric 22-sat constellation.

(b) BILP 18-sat constellation.

Figure 2.11: Example 1: 3D view of generated constellations at $n = 0$ (ECI frame).

a specialized threshold function, namely, a square wave function:

$$f[n] = \begin{cases} 2, & \text{for } 240 \leq n \leq 480 \\ 1, & \text{otherwise} \end{cases}$$

A coverage requirement is now time-dependent; the value of the square wave function varies between values 1 and 2. This requires that some parts of the simulation period must be continuously covered by at least two satellites (double-fold) and by at least one satellite

(single-fold) during the other part of the simulation period. This case is an abstract illustration of general time-varying constellation applications. For example, a communication satellite constellation may require two satellites during the day for doubled-capacity and one satellite during the night for a quiescent mode.

The results are obtained as follows:

$$x^*_{\text{qs}}[n] = \begin{cases} 1, & \text{for } n = \begin{array}{l} 0, 22, 44, 65, 87, 109, 131, 153, 175, 196, 218, 240, 262, 284, \ldots \\ 305, 327, 349, 371, 393, 415, 436, 458, 480, 502, 524, 545, \ldots \\ 567, 589, 611, 633, 655, 676, 698 \end{array} \\ 0, & \text{otherwise} \end{cases}$$

$$x^*_{\text{bilp}}[n] = \begin{cases} 1, & \text{for } n = \begin{array}{l} 5, 23, 39, 75, 89, 114, 124, 130, 164, 215, 230, 255, 265, 483, \ldots \\ 493, 518, 533, 584, 618, 624, 634, 659, 673, 709 \end{array} \\ 0, & \text{otherwise} \end{cases}$$

where the total number of satellites obtained for each method is $N_{\text{qs}} = 33$ and $N_{\text{bilp}} = 24$, and the computational time is 0.1 s for the quasi-symmetric method and 3712.0 s for the BILP method. Like in Example 1, although the BILP method takes longer computational time, it can achieve a constellation pattern solution that requires a significantly smaller number of satellites than the baseline quasi-symmetric method. The distribution of satellites in the (Ω, M)-space is shown in Figure 2.12.

As shown in Figure 2.13, the BILP constellation produces a coverage timeline that closely follows the time-varying coverage requirement. Such a coverage timeline is possible since the BILP constellation is not subject to symmetry in the satellite distribution. This is not the case for the quasi-symmetric method due to its (quasi-)symmetrical satellite distribution, which resulted in a conservative solution that provides a double-fold coverage over the entire period, even when it is not needed. This leads to the superior solution from the BILP method compared with the baseline quasi-symmetric method. As observed in Ex-

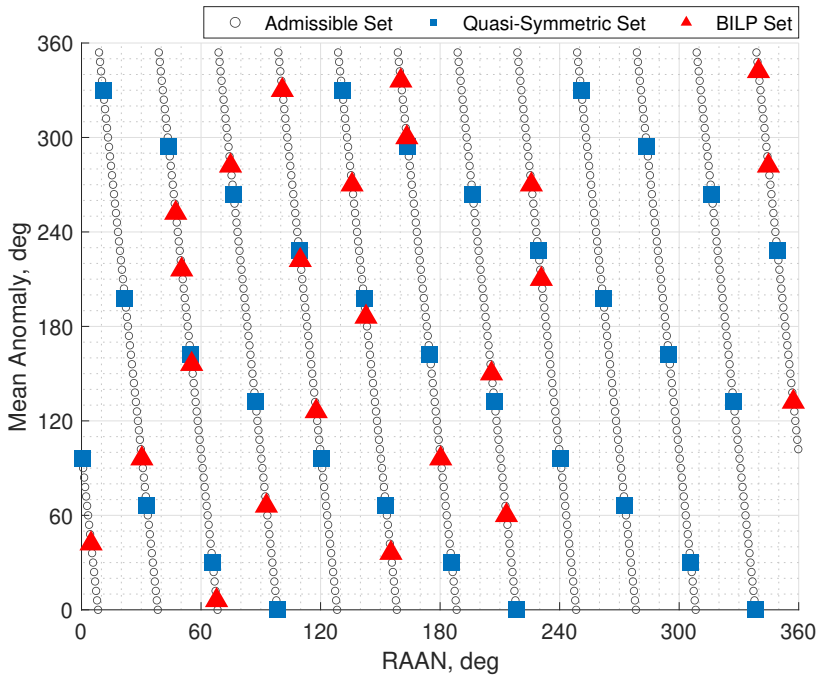
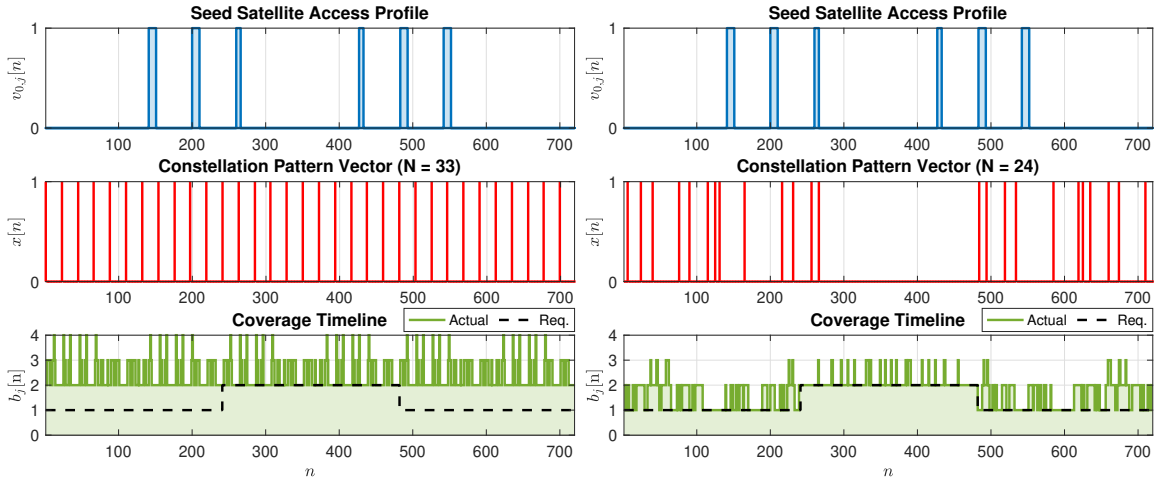


Figure 2.12: Example 2: admissible set, quasi-symmetric set, and binary integer linear programming set in the (Ω, M)-space.

ample 1, the BILP method already reduces the number of satellites required compared to that of the quasi-symmetric method given the single-fold coverage requirement. Changing only the coverage requirement to be time-varying, we further observe the additional reduction of the number of satellites for the BILP method. A snapshot of the corresponding constellation configurations at $n = 0$ is shown in Figure 2.14.

2.6.3 Example 3. Single-Fold Continuous Coverage over Multiple Target Points

For this example, we consider a target area, Antarctica, which calls for continuous and reliable telecommunication systems to support existing and planned scientific expeditions [38]. The area is discretized into a set of 94 target points following the 3°-by-3° resolution (latitude-by-longitude). All target points set $\varepsilon_{\min} = 30°$. A seed satellite orbital element vector $\mathbf{oe}_0 = [5/1, 0.41, 63.435°, 90°, 0°, 0°]^T$ (critically-inclined elliptic orbit with the apogee over the southern hemisphere) is assumed. The period of repetition is 86 076 s. The length of vectors is selected, $L = 718$, such that the time step is approximately



(a) Quasi-Symmetric method. (b) BILP method.

Figure 2.13: Example 2: the APC decomposition.

$t_{\text{step}} \approx 120\,\text{s}$. The objective of this example is to design a satellite constellation configuration that achieves single-fold continuous coverage ($\boldsymbol{f} = \mathbf{1}$) over all target points.

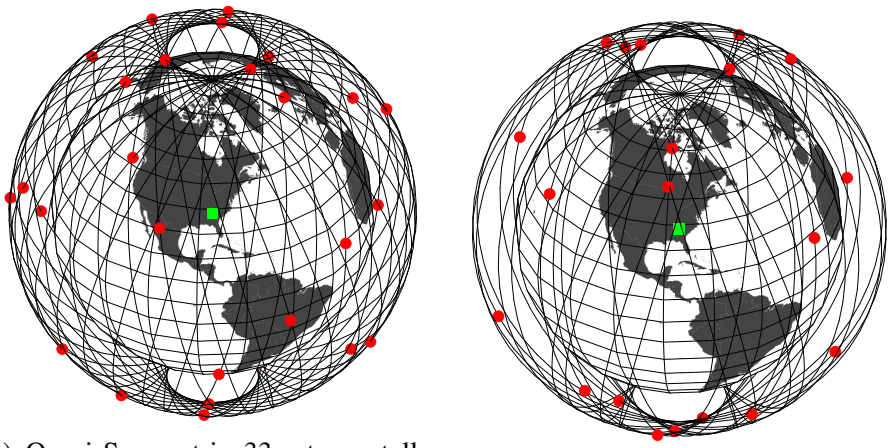
Note that this continuous polar coverage is a typical example that is often handled with a symmetric constellation, and thus we would expect that the quasi-symmetric method would perform well.

The results are obtained as follows:

$$x^*_{\text{qs}}[n] = \begin{cases} 1, & \text{for } n = 0, 120, 239, 359, 479, 598 \\ 0, & \text{otherwise} \end{cases}$$

$$x^*_{\text{bilp}}[n] = \begin{cases} 1, & \text{for } n = 96, 310, 358, 562, 612 \\ 0, & \text{otherwise} \end{cases}$$

where the total number of satellites obtained for each method is $N_{\text{qs}} = 6$ and $N_{\text{bilp}} = 5$, and the computational cost was 10.7 s for the quasi-symmetric method and 748.4 s for the BILP method. It is worth mentioning that, even for this polar-coverage example for which we would typically just use a symmetric constellation pattern (i.e., using the baseline method), the BILP method still achieves an asymmetric constellation pattern with fewer satellites. A



(a) Quasi-Symmetric 33-sat constellation.

(b) BILP 24-sat constellation.

Figure 2.14: Example 2: 3D view of generated constellations at $n = 0$ (ECI frame).

snapshot of the obtained constellations is indicated in Figure 2.16.

2.6.4 Example 4. Time-Varying and Spatially-Varying Coverage over Multiple Target Points

In this example, we design a satellite constellation system that performs remote sensing tasks over two areas of interest: the Amazon and Nile river basins. These areas represent two of the major river basins in the world thereby making them desirable locations for monitoring forests, logging, soil and water managements [40, 41], and thus are of great interest to the international community. Each area of interest is discretized into a set of target points following the 3°-by-3° resolution (latitude-by-longitude). The Amazon river basin target point set $\mathcal{J}_1$ is composed of 56 target points and the Nile river basin target point set $\mathcal{J}_2$ is composed of 30 target points. The target points are shown in Figure 2.17.

Each target point set is assumed to require different revisit time requirements: the Amazon basin has a revisit time requirement of every twelve hours, starting six hours after the epoch, whereas the Nile basin has a revisit time requirement of every six hours, starting at the epoch. We assume that all target points within the same set require simultaneous access to the system satellites at given revisit time requirements. Note that these requirements

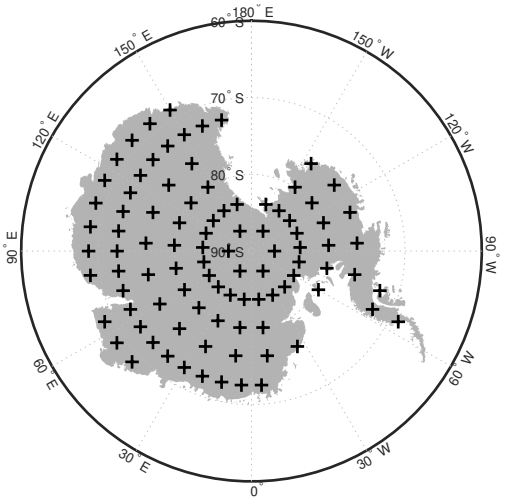


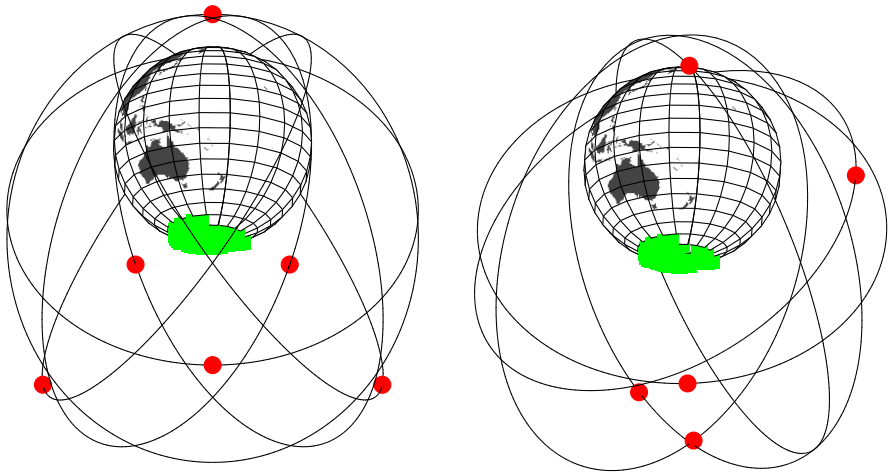
Figure 2.15: Example 3: Antarctica target points (3°-by-3° resolution); the shapefile is obtained from Ref. [39].

are not just constraining the revisit time interval but the exact time step for revisit; this is referred to as the *strict* revisit time requirement here. Furthermore, all target points are assumed to require the minimum elevation angle threshold of 20°, which corresponds to the hypothetical sensor's field-of-view of approximately 110° at a given altitude of satellites. The length of vectors is chosen, $L = 4200$ ($t_{\text{step}} \approx 123.4\,\text{s}$), such that we can represent the complex coverage requirements in an integer-indexed symmetrical form. Note that this coverage requirement is both time-varying (i.e., periodic) and spatially-varying (i.e., different requirements for Amazon and Nile river basin target points).

$$f_j[n] = \begin{cases} 1, & \text{for } n = 175, 525, 875, ..., 4025 \\ 0, & \text{otherwise} \end{cases} \qquad \forall j \in \mathcal{J}_1$$

$$f_j[n] = \begin{cases} 1, & \text{for } n = 0, 175, 350, 525, 700, 875, ..., 4025 \\ 0, & \text{otherwise} \end{cases} \qquad \forall j \in \mathcal{J}_2$$

A single-subconstellation system is assumed with the corresponding seed satellite orbital elements vector: $\mathbf{oe}_0 = [83/6, 0, 99.2°, 0°, 0°, 0°]^T$. This orbit corresponds to an alti-



(a) Quasi-Symmetric 6-sat constellation.

(b) BILP 5-sat constellation.

Figure 2.16: Example 3: 3D view of generated constellations at $n = 0$ (ECI frame).

tude of 946.7 km. The period of repetition of this orbit is $T_r = 5.184 \times 10^5$ s, which is six days. The system must satisfy:

$$\begin{bmatrix} \boldsymbol{V}_{0,1} \\ \boldsymbol{V}_{0,2} \\ \vdots \\ \boldsymbol{V}_{0,86} \end{bmatrix} \boldsymbol{x} \geq \begin{bmatrix} \boldsymbol{f}_1 \\ \boldsymbol{f}_2 \\ \vdots \\ \boldsymbol{f}_{86} \end{bmatrix}$$

where the dimension of this inequality is $(361200 \times 4200) \cdot (4200 \times 1) \geq (361200 \times 1)$.

The results show that the quasi-symmetric constellation is composed of 96 satellites, whereas the BILP constellation is composed of 29 satellites. Comparing the computational cost, the quasi-symmetric method took 486.7 s, whereas the BILP method took only 7.7 s. This shows a significant improvement of the BILP method in terms of both the number of satellites and the computational time with respect to the quasi-symmetric constellation. The quasi-symmetric method is performing poorly because we need a large number of satellites if the symmetric pattern is used. This factor, together with the large numbers of target points and time steps, makes the iterative process in the quasi-symmetric method inefficient. The BILP method, instead, identifies the asymmetric optimal solution with a significantly smaller number of satellites. The low computational cost for the BILP method

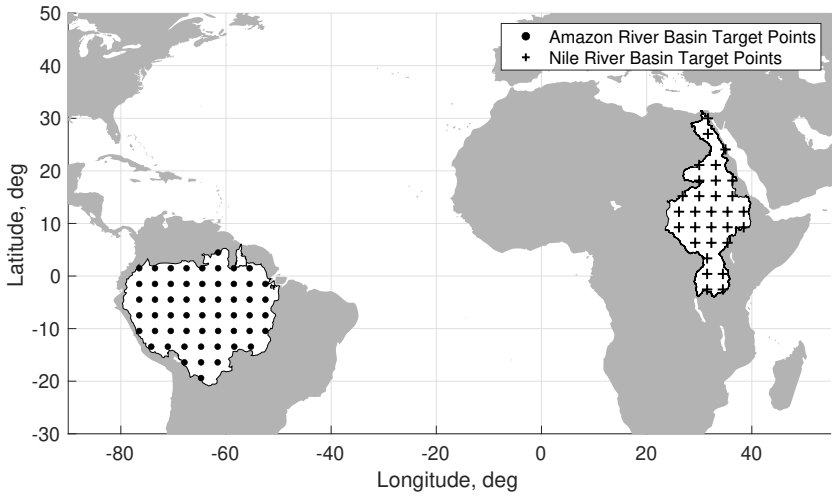
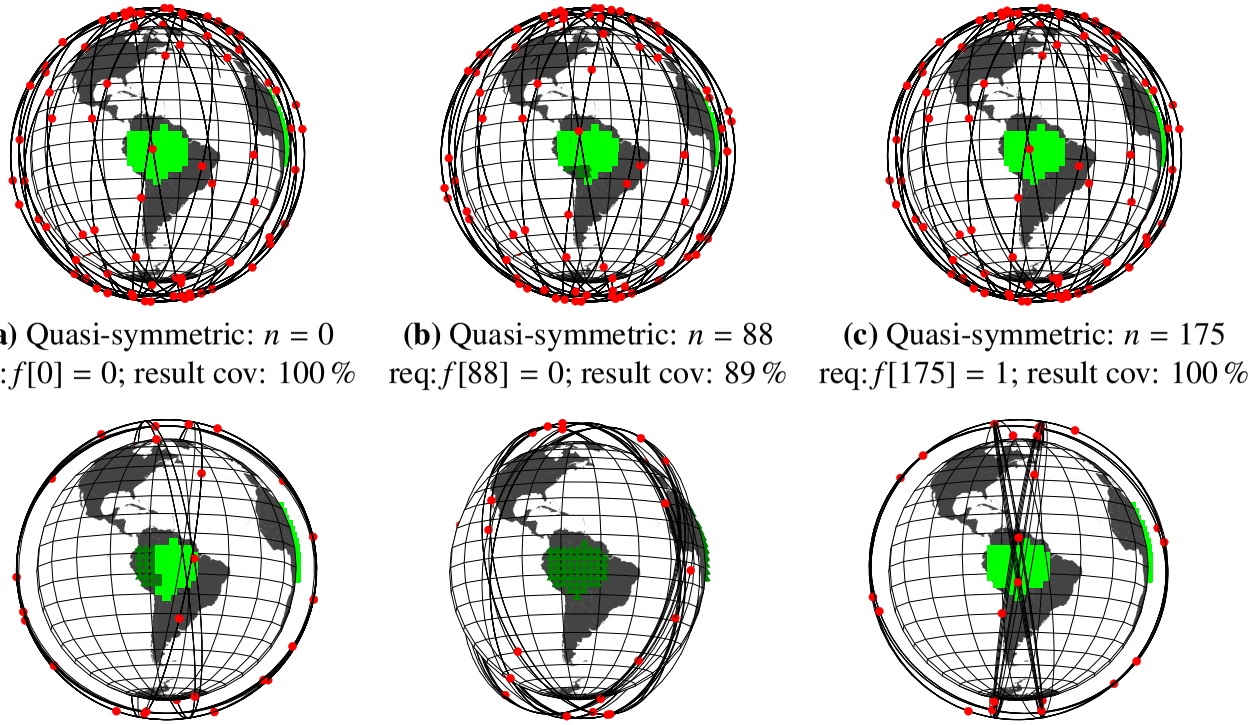


Figure 2.17: Example 4: Amazon and Nile river basin target points (3°-by-3° resolution); the polygon shapefiles are retrieved from the dataset provided by the World Bank [42].

is due to the BILP solver, Gurobi in our case; the problem structure allows Gurobi to perform an efficient presolve procedure, resulting in a short optimization time.

Figure 2.18 and Figure 2.19 show the select snapshots of both the quasi-symmetric constellation and the BILP constellation in chronological order over the Amazon and Nile river basins, respectively. (Due to the large number of satellites, the resulting constellation pattern vector is omitted.) As expected, both constellations provide simultaneous access to the target points when needed ($n = 175$ for Amazon river basin; $n = 0, 175$ for the Nile river basin), satisfying the strict revisit time requirements. It can be seen that, while the quasi-symmetric method satisfies the coverage requirements with a (quasi-)symmetric constellation pattern, the BILP method takes advantage of the asymmetry and satisfies the same requirements with fewer satellites.



(a) Quasi-symmetric: $n = 0$
req:$f[0] = 0$; result cov: 100 %

(b) Quasi-symmetric: $n = 88$
req:$f[88] = 0$; result cov: 89 %

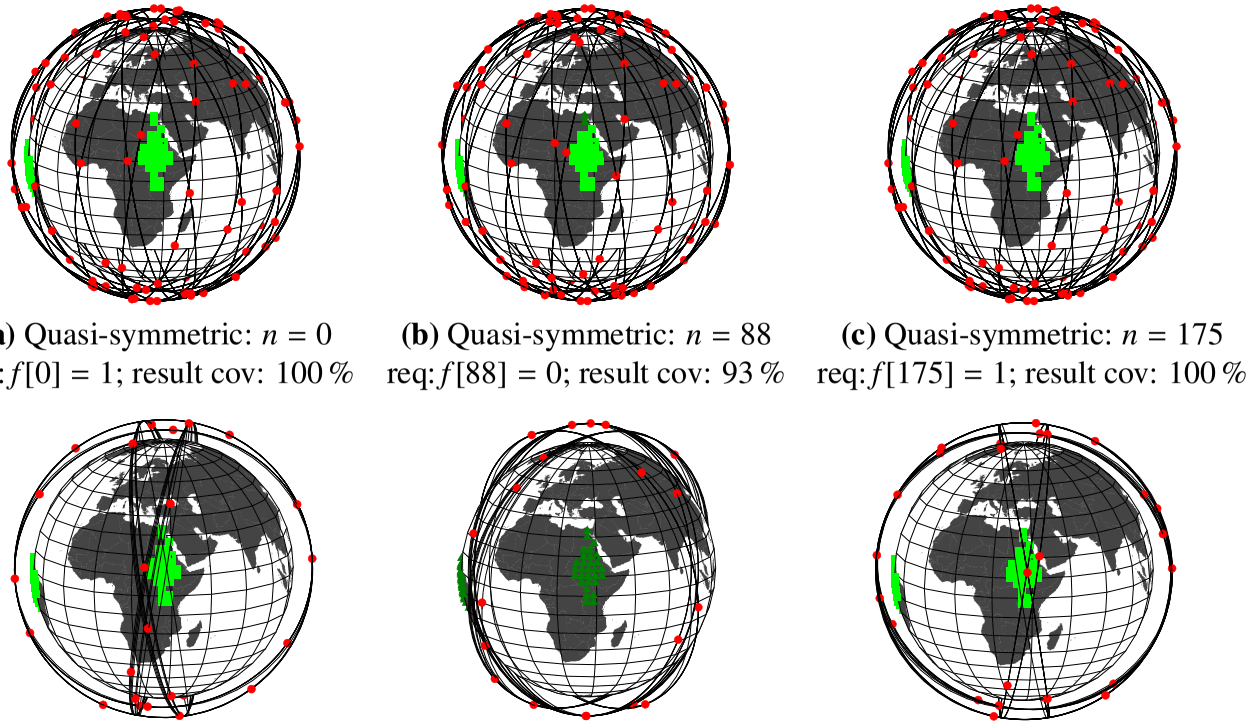
(c) Quasi-symmetric: $n = 175$
req:$f[175] = 1$; result cov: 100 %

(d) BILP: $n = 0$
req:$f[0] = 0$; result cov: 66 %

(e) BILP: $n = 88$
req:$f[88] = 0$; result cov: 0 %

(f) BILP: $n = 175$
req:$f[175] = 1$; result cov: 100 %

Figure 2.18: Example 4: Coverage over the Amazon river basin; select snapshots are shown at $n = 0, 88, 175$ (ECI frame). (a), (b), and (c) are the snapshots of the quasi-symmetric constellation and (d), (e), and (f) are the snapshots of the BILP constellation. At each n, targets that have satellite visibility are shown in light green squares and targets that do not have satellite visibility are shown in dark green triangles. "req" indicates the coverage requirement, and "result cov" is the actual coverage performance of the solution. For example, when the requirement $f[n] = 1$, the coverage has to be 100% (i.e., at least one satellite is visible from all target points in the area). It can be seen that the BILP method takes advantage of asymmetry and satisfies the coverage requirements with fewer satellites.



(a) Quasi-symmetric: $n = 0$
req:$f[0] = 1$; result cov: 100 %

(b) Quasi-symmetric: $n = 88$
req:$f[88] = 0$; result cov: 93 %

(c) Quasi-symmetric: $n = 175$
req:$f[175] = 1$; result cov: 100 %

(d) BILP: $n = 0$
req:$f[0] = 1$; result cov: 100 %

(e) BILP: $n = 88$
req:$f[88] = 0$; result cov: 0 %

(f) BILP: $n = 175$
req:$f[175] = 1$; result cov: 100 %

Figure 2.19: Example 4: Coverage over the Nile river basin; select snapshots are shown at $n = 0, 88, 175$ (ECI frame). (a), (b), and (c) are the snapshots of the quasi-symmetric constellation and (d), (e), and (f) are the snapshots of the BILP constellation. At each n, targets that have satellite visibility are shown in light green squares and targets that do not have satellite visibility are shown in dark green triangles. "req" indicates the coverage requirement, and "result cov" is the actual coverage performance of the solution. For example, when the requirement $f[n] = 1$, the coverage has to be 100% (i.e., at least one satellite is visible from all target points in the area). It can be seen that the BILP method takes advantage of asymmetry and satisfies the coverage requirements with fewer satellites.

2.6.5 Example 5. A System of Multiple Sub-Constellations over Multiple Target Points

We consider a most general case that only the BILP method can solve: a system of multiple sub-constellations over multiple target points. In this example, two target points are considered in the target point set $\mathcal{J} = \{(\phi = 64.14°\text{N}, \lambda = 21.94°\text{W}), (\phi = 19.07°\text{N}, \lambda = 72.87°\text{E})\}$: Reykjavík, Iceland ($j = 1$) and Mumbai, India ($j = 2$). The minimum elevation angle for each target point is: $\varepsilon_{1,\min} = 15°$ and $\varepsilon_{2,\min} = 10°$. The objective is to achieve single-fold continuous coverage over all target points ($\boldsymbol{f}_j = \mathbf{1},\ \forall j \in \mathcal{J}$).

Two sub-constellations are considered: $\mathbf{\alpha}_0^{(1)} = [8/1, 0, 70°, 0°, 0°, 0°]^T$ (an altitude of 4149.2 km) and $\mathbf{\alpha}_0^{(2)} = [6/1, 0, 47.915°, 0°, 0°, 0°]^T$ (an altitude of 6380.3 km). The length of vectors is selected, $L = 717$, such that the time step is approximately $t_{\text{step}} \approx 120$ s. The period of repetitions for these sub-constellations are identical, $T_{\text{r}}^{(1)} = T_{\text{r}}^{(2)} \approx 86\,024$ s, hence making two sub-constellations synchronous. Note that, even though we are using two sub-constellations for disconnected regions of interest, the sub-constellations are not defined one per region of interest; instead, they are used together to satisfy both demands in an optimal way. The goal of the BILP method is to optimize $\boldsymbol{x}^{(1)}$ and $\boldsymbol{x}^{(2)}$ concurrently such that the system satisfies the augmented linear condition:

$$\begin{bmatrix} \boldsymbol{V}_{0,1}^{(1)} & \boldsymbol{V}_{0,1}^{(2)} \\ \boldsymbol{V}_{0,2}^{(1)} & \boldsymbol{V}_{0,2}^{(2)} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}^{(1)} \\ \boldsymbol{x}^{(2)} \end{bmatrix} \geq \begin{bmatrix} \boldsymbol{f}_1 \\ \boldsymbol{f}_2 \end{bmatrix} \Leftrightarrow \{\boldsymbol{V}_{0,1}^{(1)}\boldsymbol{x}^{(1)} + \boldsymbol{V}_{0,1}^{(2)}\boldsymbol{x}^{(2)} \geq \boldsymbol{f}_1, \boldsymbol{V}_{0,2}^{(1)}\boldsymbol{x}^{(1)} + \boldsymbol{V}_{0,2}^{(2)}\boldsymbol{x}^{(2)} \geq \boldsymbol{f}_2\}$$

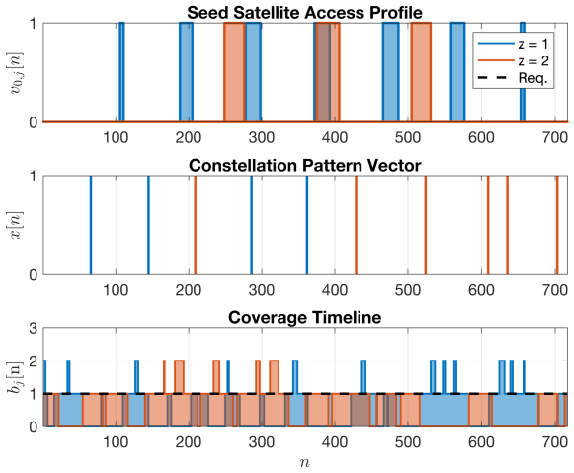
The following optimal constellation pattern vectors are obtained:

$$x^{(1)*}[n] = \begin{cases} 1, & \text{for } n = 65, 144, 285, 361 \\ 0, & \text{otherwise} \end{cases}$$

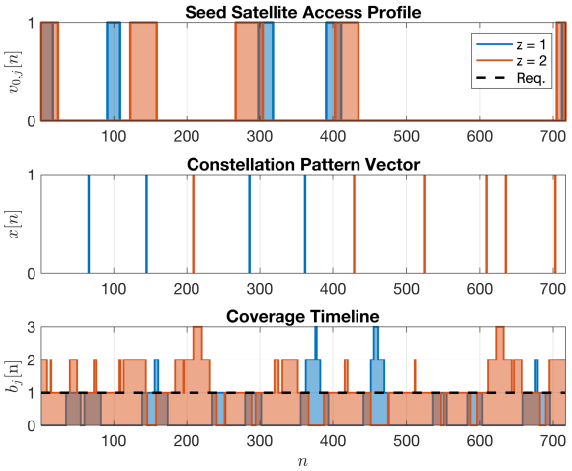
$$x^{(2)*}[n] = \begin{cases} 1, & \text{for } n = 208, 428, 523, 608, 634, 702 \\ 0, & \text{otherwise} \end{cases}$$

The number of satellites is 4 for the first sub-constellation and 6 for the second; 10 in total. The computational time was 5298.7 s.

Figure 2.20 illustrates the benefit of the BILP method. Individually, $z = 1$ sub-constellation provides 53.7 % and 37.1 % coverage over $j = 1$ and $j = 2$, respectively and $z = 2$ sub-constellation provides 65.0 % and 87.0 % coverage over $j = 1$ and $j = 2$, respectively. No individual sub-constellation alone provides complete continuous coverage over any target point. The BILP method concurrently optimizes $\boldsymbol{x}^{(1)}$ and $\boldsymbol{x}^{(2)}$ such that the continuous coverage over the whole target set $\mathcal{J}$ is achieved while minimizing the total number of satellites from two sub-constellations. Note that the constellation pattern vectors, $\boldsymbol{x}^{(1)*}$ and $\boldsymbol{x}^{(2)*}$, are identical in both sub-figures of Figure 2.20.



(a) Individual contribution over Reykjavík, Iceland.



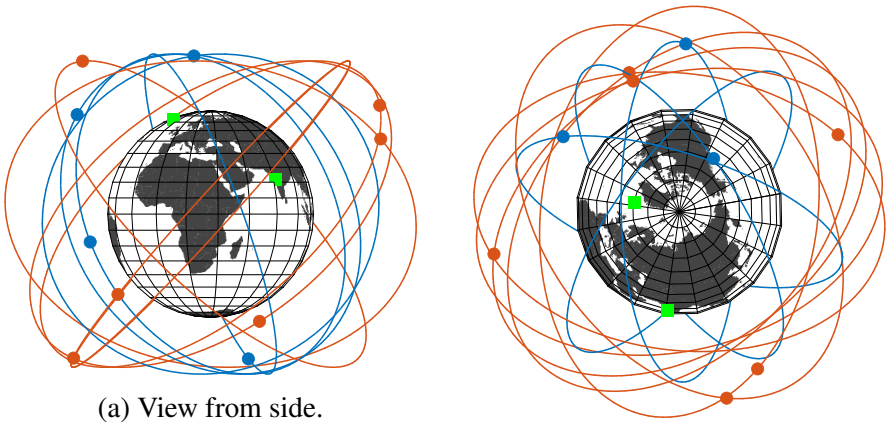
(b) Individual contribution over Mumbai, India.

Figure 2.20: Example 5: the APC decomposition.

The optimized two-subconstellation system is shown in Figure 2.21. The sub-constellation ($z = 1$) colored in blue (lower altitude) is composed of four satellites while the sub-constellation ($z = 2$) colored in red (higher altitude) is composed of six satellites for a total of ten satellites.

Finally, to show the effectiveness of having the sub-constellations, corner cases are evaluated considering each individual sub-constellation separately. The results indicate

that, under the same setting, using only the sub-constellation 1 results in 11 satellites, and using only sub-constellation 2 also results in 11 satellites. This particular case demonstrates that through the use of multiple sub-constellations, one can reduce the minimum satellites required from 11 to 10 by enlarging the design space. Also, it is worth mentioning that the BILP method can still lead to an optimal solution even for the cases where only part of the sub-constellation sets is used in the optimal pattern.



(a) View from side.

(b) View from the North Pole.

Figure 2.21: Example 5: 3D view of generated constellation at $n = 0$ (ECI frame).

2.7 Conclusions

A semi-analytical approach to optimally design a regional coverage satellite constellation pattern is proposed. By treating the seed satellite access profile and the constellation pattern vector as discrete-time signals, a circular convolution between them creates the coverage timeline. We refer to this formulation as the APC decomposition of the satellite constellation system. This formulation is used to derive a set of satellite constellation pattern design methods that take a seed satellite access profile and a coverage requirement as their inputs and output the minimum number of satellites required to satisfy the coverage requirement. Two satellite constellation pattern design methods are introduced: the baseline quasi-symmetric method and the more general BILP method. The baseline quasi-symmetric method enforces the conventional assumption of symmetry in the constellation

pattern and solves for the minimum number of satellites required in the system by incrementally increasing N until the coverage requirement is satisfied. In contrast, the new and more general BILP method solves for constellation pattern vector $\boldsymbol{x}$ where N and their temporal locations can be deduced by solving a binary integer linear programming problem. Our analysis shows that, while the quasi-symmetric method can be efficient when we can satisfy the coverage requirements with a small number of satellites in a symmetric pattern (e.g., continuous polar coverage), the BILP method always outputs optimal satellite constellation patterns that the baseline method may miss. Furthermore, the BILP method is applicable to the problems that the quasi-symmetric method cannot solve (e.g., the case with multiple sub-constellations).

Our ideas respond to the several design features that can reinforce the utility of regional constellations: multiple target points, complex time-varying and spatially-varying requirements, and multiple sub-constellations. The developed circular convolution formulation allows linearity in both the multiple target points direction and multiple sub-constellations direction via matrix augmentation. A user can design (1) a single constellation system that simultaneously satisfies the complex coverage requirement of area targets composed of multiple target points, (2) a system of multiple sub-constellations that satisfies the complex coverage requirement of a single target point, or (3) a combination of both. These design features are demonstrated via a series of illustrative examples in Section 2.6. The resulting general constellation pattern design approach can be integrated with existing orbital characteristics design methods and launch/mission constraints to help future satellite constellation designers rigorously achieve optimal constellation designs.

Despite the demonstrated effectiveness of the proposed approach, there are some possible directions for future work to improve it further. The first potential direction is related to the computational time. Due to the nature of the discretization, obtaining a high-fidelity solution computed with fine time discretization would require a large-sized problem and thus a long computational time. To make the method computationally more scalable, approxi-

mation algorithms or heuristics methods can be developed to retrieve feasible, yet potentially suboptimal, solutions in a relatively short amount of time. Furthermore, the proposed method only considers the J_2 effect as the disturbance and assumes that the spacecraft has the maneuvering capability to cancel out other disturbances. This assumption is reasonable for the proposed method to be used for a high-level constellation pattern design purpose, but it can be improved for higher-fidelity modeling. Finally, this constellation pattern design method requires the seed satellite orbital elements as its input. While Appendix C shows one example process of integrating the proposed approach into the constellation design practice, further investigation can be performed to ensure an efficient and effective integration.

This chapter is based on the following publication:

H. 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.

CHAPTER 3
REGIONAL CONSTELLATION RECONFIGURATION PROBLEM: INTEGER LINEAR PROGRAMMING FORMULATION AND LAGRANGIAN HEURISTIC METHOD

3.1 Introduction

Satellite constellation systems are often subject to varying requirements and environments during their operations. Factors that contribute to varying requirements include the change in an area of interest (e.g., disaster monitoring [43], temporary reconnaissance [44], and theater situational awareness missions) and the change in required coverage (e.g., switching from intermittent coverage to single-fold continuous coverage). In addition, the systems themselves may also be forced to change due to the addition of new satellites (e.g., a staged deployment [45, 46]) or the loss of existing satellites due to failures [47] and/or end-of-life decommissions. Under such circumstances, it is logical for system operators to seek an option to "reconfigure" an existing constellation system to maximize the utility of active on-orbit assets instead of launching a whole new constellation.

We define constellation reconfiguration as a process of transforming an existing configuration into another to maintain the system in an optimal state given a set of new mission requirements [48, 49]. The design of a reconfiguration process is nontrivial and involves interdisciplinary fields of studies such as satellite constellation design theory, orbital transfer trajectory optimization, and mathematical programming to enable a robust constellation reconfiguration framework.

Of particular interest to this chapter is the topic of satellite constellation reconfiguration in the context of regional coverage, which finds its applications in Earth observations (EO). Many present-day EO satellite systems are monolithic or small-scale constellation systems

distributed in near-polar low Earth orbits (LEO), mostly in sun-synchronous orbits to leverage consistent illumination conditions. Near-polar orbits enable EO satellite systems to scan different parts of the globe in each orbit; such an attribute is ideal in detecting changes in the Earth's land cover, vegetation, and civil infrastructures. However, the long revisit time for a particular target is unsuitable for missions that require rapid adaptive mission planning and enhanced coverage such as satellite-based emergency mapping, surveillance, and reconnaissance missions, to name a few [50, 51]. Of latest attention to the EO community has been the concept of agile satellites that assume attitude control capability, which is deemed to enhance the overall system responsiveness and scheduling efficiency [52]. Recently, several works have explored the concept of maneuverable satellites in the domain of EO satellite systems as a new paradigm to bolster the system observation capacity by directly manipulating the orbits [53, 54, 43, 55]. In this chapter, we investigate the concept of reconfiguration as a means for system adaptability and responsiveness that adds a new dimension to the operation of next-generation EO satellite constellation systems.

The problem of satellite constellation reconfiguration consists of two different, yet coupled, problems—the *constellation design problem* and the *constellation transfer problem* [48, 56, 57]. The former deals with the optimal design of a (destination) constellation configuration that satisfies a set of mission requirements; the latter is concerned with the minimum-cost transportation of satellites from one configuration to another provided the knowledge of both end states. Although we may approach these two interdependent problems independently in the sequential manner (i.e., a destination configuration is first designed and followed by the optimal assignment of satellites to new orbital slots), the outcome of such an open-loop procedure may result in a suboptimal reconfiguration process as a whole [48, 57]. Without taking into account the satellite transportation aspect in design, the optimized new configuration may be too costly or, in fact, infeasible to achieve. This background motivates us to concurrently consider constellation design and transfer aspects in satellite constellation reconfiguration.

The problem of the concurrent constellation design and transfer optimization is highly complex and challenging. While it is well known that the constellation transfer problem can be formulated as an assignment problem [58], formulating the constellation design problem faces a unique mathematical programming challenge due to (i) the potentially needed complex (time-varying) regional coverage and (ii) the reconfiguration problem with the cardinality constraint. First, the constellation design problem for complex regional coverage may need to incorporate design attributes such as heterogeneity among member satellites (e.g., different orbits and hardware specs) and asymmetry in satellite distributions. The classical constellation patterns such as the streets-of-coverage [9, 10], Walker patterns [6, 7, 8], and the tetrahedron elliptical constellation [14] are limiting due to symmetry and sparsity in satellite distribution, especially with a small number of satellites. Ref. [59] has shown that, for complex regional coverage, relaxing symmetry and homogeneity assumptions of the classical methods enable the exploration of larger design space and hence lead to the discovery of more efficient constellation pattern sets. How to incorporate the asymmetric patterns into the constellation reconfiguration while considering the transfer cost is a challenging problem that needs to be addressed. Second, in the context of satellite constellation reconfiguration, the design of a destination configuration may be restricted to a given number of satellites, which we refer to as the cardinality constraint. This is a logical assumption to make because, without the enforcement of the cardinality constraint, the optimal design of a destination configuration may require substantially more satellites than what is readily available for orbital maneuvers. Launching a set of new satellites to fulfill the deficit within a limited time window can be challenging both financially and operationally. Thus, considering the satellite reconfiguration problem with the cardinality constraint is also a challenge.

The challenge not only resides in the integration of constellation design and transfer aspects of the reconfiguration problem but also in the solution approach. A solution method that is computationally efficient and yields high-quality solutions is most desirable, es-

pecially for mission scenarios that call for rapid system response. How we formulate the design-transfer model dictates its mathematical property and the pool of applicable solution algorithms, and hence the time complexity in retrieving solutions. Several satellite constellation reconfiguration studies have been investigated encompassing both the constellation design and the constellation transfer in various problem settings [47, 44, 53, 43, 60]. While this stream of research demonstrated the value of concurrent optimization, their mathematical problem formulations were generally nonlinear and often employed meta-heuristic algorithms. Although meta-heuristic algorithms can be efficient in obtaining high-quality solutions, they can be computationally expensive for highly-constrained problems and cannot certify the optimality (or the optimality gap) of the obtained solutions. Therefore, the principal challenge we face in this work is the entire streamline of mathematical modeling and optimization that stems from the formulation of the constellation design problem to the integration of constellation design and transfer aspects and the development of a solution method.

The main contribution of this chapter is three-fold:

(1) Constellation design problem formulation (Section 3.3). We propose a novel formulation called the *maximum coverage problem* to design a maximal regional coverage constellation configuration with the cardinality constraint. The formulation features both heterogeneity in satellite specifications (e.g., different orbits, fuel states, and sensor profiles) and asymmetry in constellation patterns. Moreover, the targets of interest can be of any nature: a point target, an area, disjointed. Also, each target point can be associated with different time-varying reward for coverage.

(2) Integrated design-transfer reconfiguration model (Section 3.4). We present an integer linear program (ILP) formulation of the design-transfer problem designated the *regional coverage constellation reconfiguration problem* (RCRP). The RCRP formulation incorporates constellation design and constellation transfer aspects that are otherwise independent and serial in the state-of-the-art [48]. The concurrent consideration of these two problems

enables the exploration of larger design space and aid operators with the trade-off analysis between the transportation cost and the coverage performance. The model supports various mission concepts of operations that arise in the context of regional coverage missions.

(3) Lagrangian heuristic method for solving RCRP (Section 3.5). We develop a dedicated computationally-efficient solution method for the proposed RCRP. The RCRP formulation enables the use of mixed-integer linear programming (MILP) methods such as the branch-and-bound algorithm to obtain provably-optimal solutions. However, the problem becomes intractable even for moderately-sized instances (e.g., ten satellites, a thousand orbital slots, and thirty target points). Responding to this challenge, we propose a Lagrangian relaxation-based heuristic method to approach large-scale optimization. The key idea is to relax a set of complicating constraints to reveal and exploit the special substructure of the problem that is much easier to solve. The results of the computational experiments attest to the near-optimality of the Lagrangian heuristic solutions compared to the solutions obtained by a commercial solver at significantly faster runtime.

The remainder of this chapter is organized as follows. Section 3.2 overviews the background materials. Section 3.3 introduces the constellation design problem with the cardinality constraint. Section 3.4 provides a mathematical formulation of the design-transfer problem and discusses the characteristics of the problem. Then, Section 3.5 introduces the developed Lagrangian heuristic method to solve the proposed problem formulation. Section 3.6 introduces a possible extension of the proposed formulation and method that can be practically important. In Section 3.7, we conduct computational experiments to demonstrate the value of the developed method and provide an illustrative example applied to the case of federated disaster monitoring. Finally, Section 3.8 concludes this chapter.

3.2 Preliminaries

The primary objectives of this chapter are to (i) propose an optimization problem formulation that models the constellation design problem, (ii) construct an integrated constellation design-transfer reconfiguration model, and (iii) develop a dedicated solution method. In this section, we discuss materials that are relevant to (i) and (ii). In Section 3.2.1, we review the constellation-coverage model that serves as a building block for the mathematical formulation of the constellation design problem. In Section 3.2.3, we review a constellation transfer problem model by Ref. [58] focusing on the formulation and the mathematical property.

Notation. We use boldface letters to represent vectors, for example, $\boldsymbol{\varphi} = (\varphi_{ijs} \in \{0,1\} : i \in \mathcal{I}, j \in \mathcal{J}_s, s \in \mathcal{S})$ and $\boldsymbol{y} = (y_{tp} \in \{0,1\} : t \in \mathcal{T}, p \in \mathcal{P})$. We use the asterisk symbol in superscript $(\cdot)^*$ to denote the optimality of a variable $(\cdot)$. We denote $Z(\cdot)$ the optimal objective function value of a given problem with parameters $(\cdot)$. Z_{LP} denotes the optimal value of a given problem with integrality constraints dropped, hence the name linear programming (LP) relaxation bound. $\text{Co}(\cdot)$ denotes the convex hull of a set $(\cdot)$. We let $|\cdot|$ denote the cardinality of a set $(\cdot)$.

3.2.1 Constellation-Coverage Model

We introduce the constellation-coverage model that relates the configuration of a constellation system with its coverage performance. In this model, the finite time horizon of period T is discretized into a set of time steps with time step size Δt. Let $\mathcal{T} \coloneqq \{0, 1, \ldots, m-1\}$ (where $m\Delta t = T$) be the set of time step indices t such that the set $\{t(\Delta t) : t \in \mathcal{T}\}$ is the discrete-time finite horizon. Let $\mathcal{J}$ be the set of orbital slot indices; each orbital slot j possesses a set of unique orbital elements $\boldsymbol{\text{œ}}_j = (a_j, e_j, inc_j, \omega_j, \Omega_j, M_j)$. Here, a, e, inc, ω, Ω, and M each represents the semi-major axis, eccentricity, inclination, argument of periapsis, right ascension of ascending node (RAAN), and mean anomaly of an orbit, respectively.

We let $\mathcal{P}$ be the set of target point indices p.

Model Definitions

For the ease of description, and without loss of generality, we consider the model for a single target point.

Definition 1 (Visibility matrix)**.** Let V_{tj} denote the Boolean visibility state that equals 1 if a satellite on orbital slot j covers the target point at time step t. We let $\boldsymbol{V} = (V_{tj} \in \mathbb{Z}^2_{\geq 0} : t \in \mathcal{T}, j \in \mathcal{J})$ denote a visibility matrix where $\mathbb{Z}_{\geq 0}$ denotes the set of non-negative integers.

To construct $\boldsymbol{V}$, the following parameters need to be specified for each orbital slot: the orbital elements $\boldsymbol{\text{œ}}_j$, the field-of-view of a satellite sensor, the coordinates of a target point, and the epoch at which the finite time horizon is referenced to. With these parameters, the orbital slot is propagated under the governing equations of motion (e.g., J_2-perturbed Keplerian motion) for the finite time horizon of period T; at each time step t, the Boolean visibility masking is applied to construct an element of a visibility matrix V_{tj}.

Definition 2 (Constellation pattern vector)**.** A constellation pattern vector $\boldsymbol{x} = (x_j \in \{0, 1\} : j \in \mathcal{J})$ specifies the relative distribution of satellites in a given system (or simply, the configuration of a constellation system). Each element of $\boldsymbol{x}$ is defined as:

$$x_j \coloneqq \begin{cases} 1, & \text{if a satellite occupies orbital slot } j \\ 0, & \text{otherwise} \end{cases}$$

Definition 3 (Coverage timeline)**.** Let b_t be the number of satellite(s) in view from the target point at time step t. Then, we let $\boldsymbol{b} = (b_t \in \mathbb{Z}_{\geq 0} : t \in \mathcal{T})$ denote a coverage timeline. Here, the visibility of a satellite from a target point follows from the Boolean visibility masking.

Remark 1 (A Linear System). We can relate visibility matrix $\boldsymbol{V}$, constellation pattern vector

$\boldsymbol{x}$, and coverage timeline $\boldsymbol{b}$ as a linear system. Mathematically,

$$b_t = \sum_{j \in \mathcal{J}} V_{tj} x_j \tag{3.1}$$

In this model, the set $\mathcal{J}$ can comprise orbital slots with different orbital characteristics without any predefined rule. Because satellites on these orbital slots experience different degrees of orbital perturbations over time, the constellation-coverage model is only valid within the specified time horizon of period T. There will be a loss of fidelity in the constellation and coverage relationship beyond the specified time horizon. Such a case is indeed suitable for the planning of many temporary mission operations. However, some cases require persistency in coverage of a region of interest for a long-term horizon. To account for it, we make several assumptions about the constellation-coverage model. In what follows, we provide an overview of the special case.

Special Case: APC Decomposition

To guarantee persistent regional coverage, Ref. [59] introduced a particular constellation-coverage model called the APC decomposition (named after the three finite discrete-time sequences of the special case model: the visibility (Access) profile, constellation Pattern vector, and Coverage timeline) by making two assumptions about the constellation model: (i) the repeating ground track (RGT) orbits and (ii) the common ground track constellation. In what follows, we review the assumptions and the definitions of APC decomposition model by Ref. [59].

Assumption 1 (Repeating ground track orbit)**.** All system satellites are placed on RGT orbits.

A ground track is the trace of a satellite's sub-satellite points on the surface of a planetary body. A satellite on an RGT orbit makes N_{P} number of revolutions in N_{D} number of nodal periods. There is a finite time horizon of period T (often called a period of rep-

etition) for which a satellite repeats its closed relative trajectory exactly and periodically. Expressing this condition, we get:

$$T = N_{\mathrm{P}} T_{\mathrm{S}} = N_{\mathrm{D}} T_{\mathrm{G}}$$

where N_{P} and N_{D} are positive integers. T_{S} is the nodal period of a satellite due to both nominal motion and perturbations and T_{G} is the nodal period of Greenwich.

When designing a satellite constellation pattern for persistent regional coverage, it is important to consider the resonance of a satellite orbit with respect to a region of interest. Because an RGT orbit is resonant with the planetary body's rotation, its satellite-to-target coverage state can be modeled as a periodic finite-state automaton for a stationary target on the surface of a planetary body. This logic is well supported in a quantitative study by Hanson et al., which has shown that RGT orbits yield better partial coverage performance than non-RGT orbits [16].

Assumption 2 (Common ground track constellation)**.** All system satellites follow the same trajectory in a rotating frame of reference.

All satellites in a common ground track constellation share identical semi-major axis a, eccentricity e, inclination inc, and argument of periapsis ω but each satellite k independently holds right ascension of ascending node (RAAN) Ω_k and initial mean anomaly M_k pairs that satisfy the following distribution rule [30]:

$$N_{\mathrm{P}} \Omega_k + N_{\mathrm{D}} M_k = \text{constant mod } 2\pi$$

Lastly, we comment on the orbital characteristics for a set of candidate orbits for regional coverage. To provide a persistent regional coverage, it is important to consider the long-term stability of satellite orbits under perturbation effects. Therefore, in addition to our earlier assumption on RGT orbits, we assume a set of candidate orbits to be either circu-

lar orbits or elliptic orbits with critical inclinations so that no high orbital maintenance costs are required to negate perturbation effects (e.g., maintaining the argument of periapsis).

With these assumptions, we add following new definitions to the model to accommodate the special case.

Definition 4 (Reference visibility profile)**.** Let v_t denote the Boolean visibility state that equals 1 if a reference satellite covers a target point at time step t (0 otherwise). Then, we denote $\boldsymbol{v} = (v_t \in \{0, 1\} : t \in \mathcal{T})$ the reference visibility profile.

Definition 5 (Visibility circulant matrix)**.** A visibility circulant matrix $\boldsymbol{V}$ is the $m \times m$ matrix whose columns are the cyclic permutations of $\boldsymbol{v}$:

$$\boldsymbol{V} = \text{circ}(\boldsymbol{v}) = \begin{bmatrix} v_0 & v_{m-1} & \cdots & v_1 \\ v_1 & v_0 & \cdots & v_2 \\ \vdots & \vdots & \ddots & \vdots \\ v_{m-1} & v_{m-2} & \cdots & v_0 \end{bmatrix}$$

where the (t, j) entry of $\boldsymbol{V}$ is denoted with the modulo operator as $V_{tj} = v_{(t-j) \bmod m}$; $\text{circ}(\cdot)$ is the circulant operator that takes its argument and generates a circulant matrix as defined above.

Remark 2 (Cyclic property [59]). Under the aforementioned assumptions, a common RGT constellation system admits the *cyclic property*. The cyclic property states that a visibility profile is a cyclic shift of the reference visibility profile. This property follows from the fundamental assumptions of RGT orbits and common ground track constellations.

Remark 3 (Circular convolution operation). Following from Remark 2, we can relate reference visibility profile $\boldsymbol{v}$, constellation pattern vector $\boldsymbol{x}$, and coverage timeline $\boldsymbol{b}$ in the manner prescribed by a *circular convolution operation*. Mathematically,

$$b_t = \sum_{j \in \mathcal{J}} v_{(t-j) \bmod m} x_j \tag{3.2}$$

Following from the definition of the (t, j) entry of $\boldsymbol{V}$ in Definition 5, Eq. (3.2) can be written as a linear system in terms of a reference visibility circulant matrix: $b_t = \sum_{j \in \mathcal{J}} V_{tj} x_j$, which is in the form of Eq. (3.1).

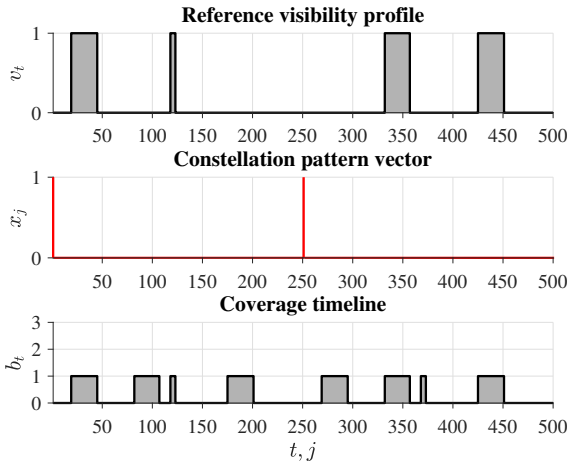
There are two notable benefits to this special case. One unique advantage of this special case is that one only requires the knowledge of the reference visibility profile and the distribution of satellite along the common relative trajectory to quantify the satellite coverage b_t of a target point at time step t. The construction of $\boldsymbol{V}$ is significantly faster than the generic case due to the use of the circulant operator. Another advantage is that, as will be discussed later in this chapter, having $\boldsymbol{V}$ as a circulant matrix can lead to useful mathematical properties. In particular, in Section 3.3, we show that the upper bound of the LP relaxation of the maximum coverage problem can be analytically computed if $\boldsymbol{V}$ is circulant. Although not directly relevant to the main contents of this chapter, circulant matrices can be used to leverage efficient solution methods for certain class of problems (e.g., set covering problems with circulant matrices in 3.2.2).

Illustrative Example

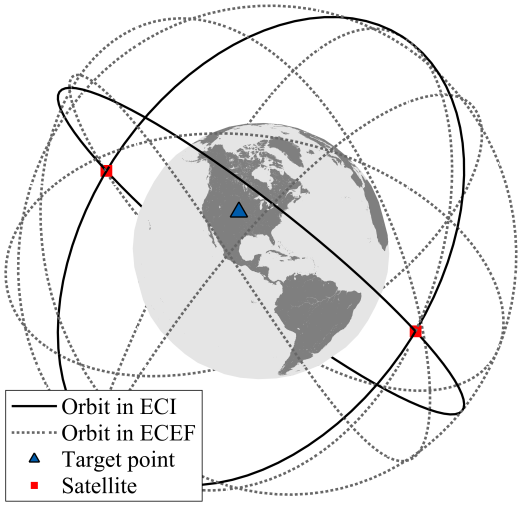
Example 1 (2-satellite system)**.** We illustrate how all definitions of the model, so called the vectors of a system, come into play in a simple example. Let $\boldsymbol{\mathfrak{oe}}_0 = (a, e, i, \Omega, u) = (12\,758.5\,\text{km}, 0, 50^\circ, 50^\circ, 0^\circ)$ be the orbital elements of the reference satellite defined in J2000 frame where u is the argument of latitude defined for circular orbits; this corresponds to the RGT ratio of $N_\text{P}/N_\text{D} = 6/1$, that is, a satellite makes six revolutions in one nodal day. Assume a single target point of interest p with the geodetic coordinate (40°N, 100°W). The minimum elevation angle threshold ϑ_min of the target is set to 10 deg. Define an arbitrary constellation pattern vector $\boldsymbol{x}$ with length $m = 500$ such that $x_j = 1$ for $j \in \{0, 250\}$ and 0 otherwise. When transcribed, this constellation pattern vector represents a constellation configuration consisting of two uniformly-distributed satellites along the common relative trajectory, each with its own unique orbital elements: $\boldsymbol{\mathfrak{oe}}_1 = (12\,758.5\,\text{km}, 0, 50^\circ, 50^\circ, 0^\circ)$

and $\mathbf{oe}_2 = (12\,758.5\,\text{km}, 0, 50°, 230°, 0°)$.

The vectors of the system—$\boldsymbol{v}$, $\boldsymbol{x}$, and $\boldsymbol{b}$—are portrayed in Figure 3.1a. The top part shows the reference visibility profile $\boldsymbol{v}$ which is constructed by propagating a hypothetical satellite with $\mathbf{oe}_0$ for the finite time horizon of period T and applying the Boolean visibility mask at each time step. The constellation pattern vector $\boldsymbol{x}$ is shown in the middle. Note that in this example, $\mathbf{oe}_0 = \mathbf{oe}_1$ because $x_0 = 1$; this indicates that satellite 1 is essentially identical to the hypothetical reference satellite. The resulting coverage timeline $\boldsymbol{b}$, which follows directly from Eq. (3.2) of the cyclic property, for this two-satellite system is shown in the bottom part. The corresponding system is visualized in Figure 3.1b in both the Earth-centered inertial (ECI) and the Earth-centered, Earth-fixed (ECEF) frames. Notice that there is a single closed trajectory in the ECEF frame as opposed to two orbital planes in the ECI frame; each dot along the orbit in the ECEF frame (i.e., the dotted curve) represents an orbital slot.



(a) Illustration of $\boldsymbol{v}$, $\boldsymbol{x}$, and $\boldsymbol{b}$.



(b) Corresponding constellation configuration in 3-D.

Figure 3.1: Two-satellite system illustrated in Example 1.

3.2.2 Regional Constellation Design Problem

In Example 1, we demonstrated the analysis of a known system using the vocabularies of the regional coverage satellite constellation model from Ref. [59]. In this appendix, we present a direct application of the model by formulating an integer linear programming problem with its decision variables being $\boldsymbol{x}$. Instead of pre-specifying it, the goal is to find the optimal constellation pattern vector $\boldsymbol{x}^*$ that meets the specified coverage requirement.

Formulation 1 (Regional constellation design problem)**.** Let $\mathcal{T} = \{0, \ldots, m-1\}$ be the set of time step indices and $\mathcal{J} = \{0, \ldots, m-1\}$ be the set of orbital slot indices. The regional constellation design problem (RCDP) is concerned with the minimum-cost design of constellation configuration while satisfying the coverage requirement $\boldsymbol{r} = (r_t \in \mathbb{Z}_{\geq 0} : t \in \mathcal{T})$ imposed on a target point of interest. RCDP is formulated as an integer linear program [59]:

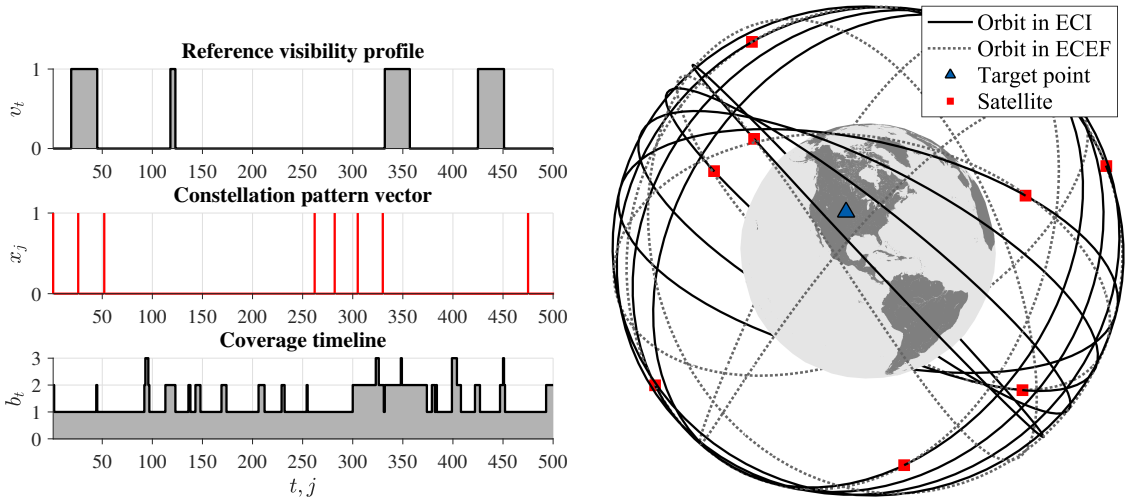
$$
\begin{aligned}
\text{(RCDP)} \quad Z = \min \quad & \sum_{j \in \mathcal{J}} c_j x_j \\
\text{s.t.} \quad & \sum_{j \in \mathcal{J}} V_{tj} x_j \geq r_t, \quad \forall t \in \mathcal{T} \\
& x_j \in \{0, 1\}, \qquad \forall j \in \mathcal{J}
\end{aligned}
$$

where the decision variable $x_j = 1$ if a satellite occupies orbit slot j ($x_j = 0$ otherwise), c_j is the cost of deploying a satellite to orbital slot j, and Z denotes the optimal value of RCDP.

The problem is an instance of the general class of set covering problems whose goal is to minimize the cost of covering all elements in the universe. Particularly, when the 0-1 integrality constraints are relaxed to the non-negative integrality constraints, the RCDP mimics the cyclic staffing problem, which is concerned with the optimal sizing and scheduling of the workforce given cyclic staffing requirements (e.g., consider a staffing requirement that

repeats weekly). The cyclic staffing problem is of significant interest to RCDP because it manifests that RCDP can be efficiently solved to optimality when the visibility profile contains a block of consecutive ones [61], which is usually observable in cases with high-altitude orbits or satellites equipped with wide field-of-view sensors. When there are multiple blocks of consecutive ones (e.g., the reference visibility profile depicted in Example 1 consists of four blocks of consecutive ones), a guaranteed-accuracy heuristic approach is available [62] with the bound: $Z_h - Z \leq \lceil (q-1)m/k \rceil$ where Z_h denotes the heuristic value, q denotes the number of blocks of consecutive ones, and k denotes the total number of ones in the reference visibility profile. Let us demonstrate RCDP in action.

Example 2 (Minimum-satellite RCDP)**.** Consider the case illustrated in Example 1. For the RCDP-specific parameters, we let $r_t = 1, \forall t \in \mathcal{T}$ (single-fold continuous coverage requirement) and $c_j = 1, \forall j \in \mathcal{J}$ (minimization of the number of satellites). Solving the RCDP to optimality, we get the optimal constellation pattern vector $\boldsymbol{x}^*$ that consists of $Z = 8$ satellites (see the middle part of Figure 3.2a). The corresponding optimal constellation configuration is pictured in Figure 3.2b.



(a) Illustration of $\boldsymbol{v}$, $\boldsymbol{x}$, and $\boldsymbol{b}$.

(b) Corresponding constellation configuration in 3-D.

Figure 3.2: RCDP solution for Example 2.

3.2.3 Constellation Transfer Problem

A constellation reconfiguration process incurs costs. In this subsection, we discuss an ILP formulation that models the constellation transfer aspect of the regional coverage constellation reconfiguration problem. Note that the notations of decision variables and sets used in this model are only valid in the context of the said problem.

Typically, the transfer problem is considered over a bipartite graph $\mathcal{G} = (\mathcal{I} \cup \mathcal{J}, \mathcal{E})$, in which the nodes of the set $\mathcal{I}$ describe the locations of the satellites while the nodes of the set $\mathcal{J}$ describe the locations of the orbital slots. Each edge $(i, j) \in \mathcal{E}$ is associated with the weight, or the cost c_{ij} of transferring satellite i to orbital slot j. In this domain, the cost commonly refers to Δv or the time-of-flight. Within this framework, the transfer problem further breaks down into two main components: (i) the combinatorial optimization to find the minimum-cost assignment of satellites from one configuration to another and (ii) the orbital transfer trajectory design between a given (satellite, orbital slot) pair. The first component is concerned with the minimum-cost bipartite matching, which can be formulated as an assignment problem [58]; see Formulation 2. The second component deals with the construction of the cost matrix by evaluating the weights of edges in the bipartite graph setting. One can quantify each weight of the edges by solving the orbital boundary value problem. Because enumerating every edge can be time-consuming, several studies proposed a rapid closed-form approximation of the true cost matrices [63, 48]. In this chapter, we limit the scope of our work to the high-thrust systems due to its benefit of timely reconfiguration, although the low-thrust systems can also be considered as an alternative option.

Formulation 2 (Assignment problem)**.** Let $\mathcal{I} = \{1, \ldots, n\}$ be the set of workers and $\mathcal{J} = \{1, \ldots, m\}$ be the set of projects. The cost of assigning worker i to project j is denoted with c_{ij}. In the case of an unbalanced (a rectangular) assignment problem (AP), the goal is to find the minimum-cost assignment of n workers to m projects such that all workers are assigned to projects, but not all projects are assigned with workers (i.e., when $n < m$). AP

can be formulated as an integer linear program:

$$
\begin{aligned}
\text{(AP)} \quad \min \quad & \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} c_{ij} \varphi_{ij} & \\
\text{s.t.} \quad & \sum_{j \in \mathcal{J}} \varphi_{ij} = 1, & \forall i \in \mathcal{I} \\
& \sum_{i \in \mathcal{I}} \varphi_{ij} \leq 1, & \forall j \in \mathcal{J} \\
& \varphi_{ij} \in \{0, 1\}, \quad & \forall i \in \mathcal{I}, \forall j \in \mathcal{J}
\end{aligned}
$$

where the decision variable $\varphi_{ij} = 1$ if worker i is assigned to project j ($\varphi_{ij} = 0$ otherwise).

We briefly discuss the concepts of totally unimodular (TU) matrices and the integral polyhedron, which are useful in leading to the discussion of the assignment problem. These concepts will come in handy later in this chapter.

Definition 6 (Total unimodularity)**.** An integral matrix $\boldsymbol{A}$ is TU if every square sub-matrix of $\boldsymbol{A}$ has determinant equal to 0, 1, or -1.

Theorem 1 (Hoffman-Kruskal [64])**.** *Let $\boldsymbol{A}$ be an integral matrix. The polyhedron $\{\boldsymbol{x} : \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}, \boldsymbol{x} \geq \boldsymbol{0}\}$ is integral for all integral vector $\boldsymbol{b}$ if and only if $\boldsymbol{A}$ is TU.*

One special feature of AP is that the problem satisfies Theorem 1. That is, the constraint matrix of AP (also known as the incidence matrix of a bipartite graph) is totally unimodular and the right-hand vector is integral; hence, the extreme points of the corresponding polytope are integral. We say that such a problem possesses the *integrality property*. Consequently, the problem can be efficiently solved as a linear program (e.g., the simplex or the interior-point methods) by relaxing the integrality constraints, also known as the linear programming relaxation, and yet obtain integral optimal solutions. Many other specialized algorithms are also available such as the polynomial-time Hungarian algorithm [65] (also known as the Kuhn-Munkres algorithm; the asymptotic complexity known to be $O(m^3)$ for

a square $m \times m$ matrix) or the auction algorithm (a pseudopolynomial complexity [66] and a polynomial complexity with ϵ-scaling [67]).

Assignment problem has attracted much attention in the field of satellite constellation reconfiguration research as an optimization model for a constellation transfer problem. Introduced in a study by de Weck et al. [58], a constellation transfer problem may be intuitively modeled as AP using the following analogy: the satellites as the workers and the orbital slots as the projects; the coefficient c_{ij} as the cost (e.g., the fuel consumption) of transferring satellite i to orbital slot j. The goal is to find the minimum-cost assignment of n satellites to m orbital slots. In this chapter, we adopt this transcription of the AP formulation in the modeling of a constellation transfer problem. For more information about the particular cost matrix generation used in this chapter, refer to Ref. [68].

3.3 Constellation Design: Maximum Coverage Problem

In this section, we introduce an ILP formulation, the maximum coverage problem (MCP), that models the constellation design aspect of the regional coverage constellation reconfiguration problem building upon the constellation-coverage model introduced in Section 3.2.1. This MCP formulation is used as a foundation for the proposed satellite constellation reconfiguration problem formulation, introduced in Section 3.4, although MCP itself is not strictly restricted to the context of constellation reconfiguration. We then illuminate on some of the interesting properties of this MCP formulation.

Consider the following problem setting. Let $\mathcal{T}$ be the set of time step indices and $\mathcal{J}$ be the set of orbital slot indices. Without loss of generality, we consider the problem for a single target point of interest. Given a finite time horizon of period T, the time-dependent observation reward $\boldsymbol{\pi} = (\pi_t \in \mathbb{R}_{\geq 0} : t \in \mathcal{T})$ is defined on a target point of interest. The goal is to locate n satellites in $\mathcal{J}$ such that the total observation reward obtained by covering the target point is maximized.

In order to obtain the reward π_t at time step t, the target point should be covered then.

We say that the target point is covered if there is at least r_t satellite(s) in view; here, the positive time-dependent coverage threshold $\boldsymbol{r} = (r_t \in \mathbb{Z}_{\geq 0} : t \in \mathcal{T})$ is a user-supplied parameter vector. We can model this system by defining two sets of decision variables—the constellation pattern variables $\boldsymbol{x}$ and coverage state variables $\boldsymbol{y} = (y_t \in \{0, 1\} : t \in \mathcal{T})$—and a set of inequalities that links $\boldsymbol{x}$ and $\boldsymbol{y}$. The decision variable $x_j = 1$ if a satellite occupies orbital slot j ($x_j = 0$ otherwise; see Definition 2). Each element y_t of the coverage state variables takes the value of unity if and only if the coverage threshold of the target point is satisfied at time step t ($y_t = 0$ otherwise). Mathematically,

$$y_t = \begin{cases} 1, & \text{if } b_t = \sum_{j \in \mathcal{J}} V_{tj} x_j \geq r_t \\ 0, & \text{otherwise} \end{cases} \tag{3.3}$$

As can be seen in Eq. (3.3), the coverage state of the target point is conditionally dependent on the configuration of a constellation system. We can linearize this relationship by introducing a set of inequalities that links $\boldsymbol{x}$ and $\boldsymbol{y}$ for all $t \in \mathcal{T}$:

$$\sum_{j \in \mathcal{J}} V_{tj} x_j \geq r_t y_t, \quad \forall t \in \mathcal{T}$$

With these conditions as constraints, MCP aims to maximize the coverage reward of a satellite configuration with a given number of satellites n. The preliminary version of the mathematical formulation of MCP is introduced in our earlier work [69], which employs the big-M method to linearize the conditional constraints. We present an improved formulation of MCP that achieves a tighter integrality gap.

Formulation 3 (Maximum coverage problem)**.** MCP is formulated as an integer linear

program:

$$\text{(MCP)} \quad Z = \max \quad \sum_{t \in \mathcal{T}} \pi_t y_t \tag{3.4a}$$

$$\text{s.t.} \quad \sum_{j \in \mathcal{J}} V_{tj} x_j \geq r_t y_t, \quad \forall t \in \mathcal{T} \tag{3.4b}$$

$$\sum_{j \in \mathcal{J}} x_j = n \tag{3.4c}$$

$$x_j \in \{0, 1\}, \qquad \forall j \in \mathcal{J} \tag{3.4d}$$

$$y_t \in \{0, 1\}, \qquad \forall t \in \mathcal{T} \tag{3.4e}$$

where Z denotes the optimal value of MCP.

The objective function (3.4a) maximizes the total reward earned by covering a given target point. Constraints (3.4b) couples the configuration of a system to its coverage state of the target point. Constraint (3.4c) is the cardinality constraint that restricts the number of satellites to a fixed value of n. Constraints (3.4d) and (3.4e) define the domains of decision variables.

The 0-1 integrality constraint on y_t can be relaxed to $0 \leq y_t \leq 1$ when $r_t = 1$. This avoids the use of unnecessary integer definitions, and could potentially facilitate the branch-and-bound alogrithm necessary to find the optimal solution [70]. Note that for $r_t > 1$, y_t may take a fractional value (take for instance, if $r_t = 2$ and $\sum_{j \in \mathcal{J}} V_{tj} x_j = 1$, then y_t can take 0.5), therefore the integrality constraints on y_t must be enforced.

Remark 4. Note that for $r_t = 1, \forall t \in \mathcal{T}$, the MCP can be shown equivalent to the maximal covering location problem (MCLP) that emerges in many problem contexts such as the facility location problem (see Figure 3.3 for an example illustration of MCLP). MCLP seeks to locate a number of facilities such that the weighted coverage of demand nodes is maximized; each facility is pre-specified with a service radius to which it can provide coverage. We have the following analogy: the satellites as the facilities and the time steps as

the demand nodes. (Conversely, this suggests the general varying-radius r-fold coverage formulation of MCLP.) Unfortunately, the equivalence in the formulations informs us that MCP is NP-hard because of the NP-hardness of MCLP [71], which can be deduced using the argument of the reduction from MCLP to MCP. For more information on the mathematical formulation and the applications of MCLP, readers are encouraged to refer to the original study by Church and ReVelle [72].

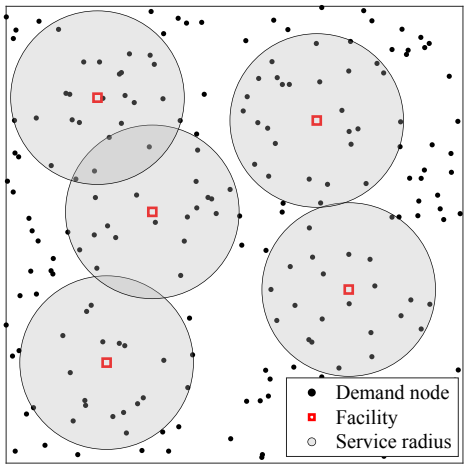


Figure 3.3: Illustration of a 5-facility MCLP; the service radius for MCP would be a visibility profile.

Expressing Z_{LP} in terms of an optimal LP solution $(\boldsymbol{x}^*, \boldsymbol{y}^*)$, we get:

$$Z_{\text{LP}} = \sum_{t \in \mathcal{T}} \pi_t y_t^* \tag{3.5}$$

where y_t^* is determined from x_j^* as

$$y_t^* = \min\left(\frac{1}{r_t} \sum_{j \in \mathcal{J}} V_{tj} x_j^*, 1\right) \tag{3.6}$$

Equation (3.6) follows from the fact that MCP is a maximization problem—y_t variables will take their maximum values as bounded by Constraints (3.4b). The second argument in the min(·) operator bounds the maximum of y_t to one, conforming with Eq. (3.3).

The discussion on the LP relaxation bound of MCP will be revisited later in this chapter

for the proposed solution method (Section 3.5), therefore we provide additional implications on Z_{LP} in this subsection for completeness. First, we notice that y_t^* in Eq. (3.5) requires the knowledge of the optimal LP solution x_j^*. However, under special conditions, we can closely approximate Z_{LP} by computing the upper bound $\hat{Z}_{\text{LP}}$ such that no knowledge of x_j^* is needed. To show this, we derive $\hat{Z}_{\text{LP}}$ from Eq. (3.5) by moving the summation inside the min(·) function. Expanding the first argument further such that the first column of $\boldsymbol{V}$ is separated, we get Eq. (3.7):

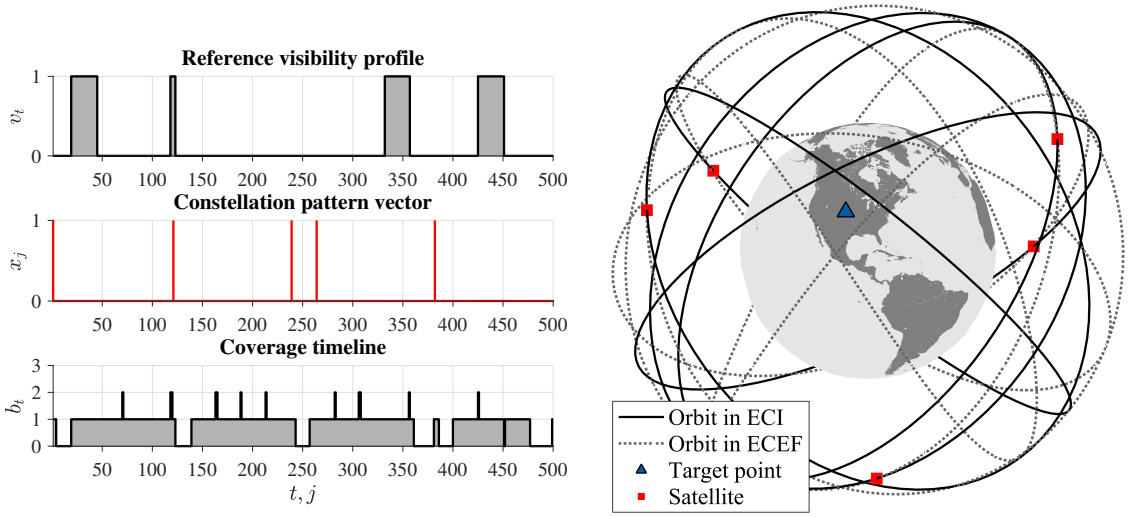
$$\begin{aligned}\hat{Z}_{\text{LP}} &:= \min\Bigg(\underbrace{n\sum_{t\in\mathcal{T}}\xi_t V_{t1}}_{(1)} + \underbrace{\sum_{j\in\mathcal{J}\setminus\{1\}} x_j^*\Bigg[\sum_{t\in\mathcal{T}}\xi_t V_{tj} - \sum_{t\in\mathcal{T}}\xi_t V_{t1}\Bigg]}_{(2)}, \sum_{t\in\mathcal{T}}\pi_t\Bigg) \\ &\geq Z_{\text{LP}}\end{aligned} \tag{3.7}$$

where $\xi_t := \pi_t / r_t$. We group the first argument of the min(·) function into two terms.

If $\xi_t = \xi, \forall t \in \mathcal{T}$ and $\boldsymbol{V}$ is a circulant matrix, then the terms within the bracket in Term (2) cancel out. Most problem instances we deal with in this chapter assume $r_t = r, \forall t \in \mathcal{T}$ (time-invariant r-fold continuous coverage) and $\pi_t = \pi, \forall t \in \mathcal{T}$ (uniform coverage reward) such that Term (2) vanishes. Therefore, we can conveniently express $\hat{Z}_{\text{LP}} = \min(n\sum_{t\in\mathcal{T}}\xi_t v_t, \sum_{t\in\mathcal{T}}\pi_t)$ as a function of known parameters n, ξ, and $\boldsymbol{v}$ and without the needing to run LP.

Example 3 (Single-fold MCP)**.** Consider the case illustrated in Example 1. Suppose we now want to maximize the coverage over the same target point p with only five satellites. For the MCP-specific parameters, we let $r_t = 1, \forall t \in \mathcal{T}$ and $\pi_t = 1, \forall t \in \mathcal{T}$ (maximization of the coverage percentage). Solving the MCP to optimality, we get the optimum of $Z = 398$, which translates into 79.6% temporal coverage of target point p by the optimal five-satellite configuration during the given repeat period T. Because $\boldsymbol{r}$ and $\boldsymbol{\pi}$ are time-invariant, we can easily approximate Z_{LP} by computing Term (1) of Eq. (3.7): we get $\hat{Z}_{\text{LP}} = 410$. In fact, by directly solving the LP relaxation problem, we obtain $Z_{\text{LP}} = 410$, which is identical

to $\hat{Z}_{\text{LP}}$. The results are visualized in Figure 3.4.



(a) Illustration of $\boldsymbol{v}$, $\boldsymbol{x}$, and $\boldsymbol{b}$.

(b) Corresponding constellation configuration in 3-D.

Figure 3.4: MCP solution for Example 3.

3.4 Regional Coverage Constellation Reconfiguration Problem

3.4.1 Problem Description

Suppose a group of heterogeneous[1] satellites undertakes a reconfiguration process to form a new configuration to maximize the observation reward of a newly emerged set of spot targets $\mathcal{P}$. Each target p in $\mathcal{P}$ is associated with the time-dependent observation reward $\pi_{tp}, \forall t \in \mathcal{T}$ where $\mathcal{T}$ is the set of time step indices. The reconfiguration process involves (i) the design of the maximum-reward destination configuration and (ii) the minimum-cost assignment of satellites between the initial and destination configurations. The goal of the problem is to identify a set of non-dominated solutions in the objective space spanned by these two competing objectives (i) and (ii).

We shall refer to this problem as the regional coverage constellation reconfiguration problem or RCRP in short. We use the vocabulary regional coverage constellations to

[1]The term heterogeneity embodies a general mission scenario of a federated system of satellites with different hardware specifications, orbital elements, and fuel states.

accentuate heterogeneity and asymmetry as distinctive design philosophies from the homogeneity and symmetry of classical global coverage constellations.

3.4.2 Mathematical Formulation

In this subsection, we propose a mathematical formulation of the regional coverage constellation reconfiguration problem. We define following sets, indices, parameters, and decision variables:

Sets and indices

$\mathcal{I}$ Set of satellite indices (index i)

$\mathcal{J}$ Set of orbital slot indices (index j)

$\mathcal{P}$ Set of target point indices (index p)

$\mathcal{S}$ Set of subconstellation indices (index s)

$\mathcal{T}$ Set of time step indices (index t)

Parameters

c_{ijs} Cost of assigning satellite i to orbital slot j of subconstellation s ($c_{ijs} \geq 0$)

π_{tp} Reward for covering target point p at time step t ($\pi_{tp} \geq 0$)

$$V_{tjps} \begin{cases} 1, & \text{if a satellite on orbital slot } j \text{ of subconstellation } s \text{ covers target point } p \\ & \text{at time step } t \\ 0, & \text{otherwise} \end{cases}$$

r_{tp} Coverage threshold of target point p at time step t ($r_{tp} \geq 1$)

Decision variables

$$\varphi_{ijs} \begin{cases} 1, & \text{if satellite } i \text{ is allocated to orbital slot } j \text{ of subconstellation } s \\ 0, & \text{otherwise} \end{cases}$$

$$y_{tp} \begin{cases} 1, & \text{if target point } p \text{ is covered at time step } t \ (b_{tp} \geq r_{tp}) \\ 0, & \text{otherwise} \end{cases}$$

We generalize the formulations to accommodate multiple target points (index p) and

multiple subconstellations (index s). The concept of *subconstellation* is applied [59]. A subconstellation is a group of satellites in a given constellation system that share common orbital characteristics such as the semi-major axis, eccentricity, inclination, and argument of periapsis. Consequently, a constellation system can therefore consist of multiple subconstellations (an example would be a multi-layered constellation). Extending this concept to orbital slots, the set of all orbital slot indices $\mathcal{J}$ can be partitioned into $|\mathcal{S}|$ subsets such that

$$\mathcal{J} = \bigcup_{s \in \mathcal{S}} \mathcal{J}_s$$

where $\mathcal{J}_s \subseteq \mathcal{J}$ denotes the set of orbital slot indices of subconstellation $s \in \mathcal{S}$, and $\mathcal{S}$ is the index set of $\mathcal{J}_s$. For the case with the RGT orbit assumption, we enforce the *synchronous condition* to guarantee identical periods of repetition for all subconstellations: $T_s = T, \forall s \in \mathcal{S}$ where T_s is the period of repetition for subconstellation s.

The mathematical formulation of RCRP is as follows

$$\text{(RCRP)} \quad \min \quad \left[\begin{array}{l} \displaystyle\sum_{s \in \mathcal{S}} \sum_{j \in \mathcal{J}_s} \sum_{i \in \mathcal{I}} c_{ijs} \varphi_{ijs} \\ \displaystyle - \sum_{p \in \mathcal{P}} \sum_{t \in \mathcal{T}} \pi_{tp} y_{tp} \end{array} \right] \tag{3.8a}$$

$$\text{s.t.} \quad \sum_{s \in \mathcal{S}} \sum_{j \in \mathcal{J}_s} \varphi_{ijs} = 1, \qquad \forall i \in \mathcal{I} \tag{3.8b}$$

$$\sum_{i \in \mathcal{I}} \varphi_{ijs} \leq 1, \qquad \forall j \in \mathcal{J}_s, \forall s \in \mathcal{S} \tag{3.8c}$$

$$\sum_{s \in \mathcal{S}} \sum_{j \in \mathcal{J}_s} \sum_{i \in \mathcal{I}} V_{tjps} \varphi_{ijs} \geq r_{tp} y_{tp}, \qquad \forall t \in \mathcal{T}, \forall p \in \mathcal{P} \tag{3.8d}$$

$$\varphi_{ijs} \in \{0, 1\}, \qquad \forall i \in \mathcal{I}, \forall j \in \mathcal{J}_s, \forall s \in \mathcal{S} \tag{3.8e}$$

$$y_{tp} \in \{0, 1\}, \qquad \forall t \in \mathcal{T}, \forall p \in \mathcal{P} \tag{3.8f}$$

RCRP is formulated as a bi-objective ILP. The first objective function in (3.8a) minimizes the total cost of a constellation reconfiguration process; the second objective func-

tion in (3.8a) maximizes the total reward earned by covering a set of target points. Constraints (3.8b) and (3.8c) are the AP-based constraints; Constraints (3.8b) ensure that every satellite is assigned to an orbital slot and Constraints (3.8c) restrict at most one satellite is to be occupied per orbital slot. Constraints (3.8d) are the MCP-based constraints; these constraints ensure that the target point p is covered at time step t only if there exists at least r_{tp} satellite(s) in view. Note that the cardinality constraint of MCP [i.e., Constraint (3.4c) of Formulation 3] is omitted in this formulation because it is implied by the satellite indices set $\mathcal{I} = \{1, \ldots, n\}$ and the AP constraints. Constraints (3.8e) and (3.8f) define the domains of decision variables.

Notice the decision variables of RCRP—they are in the form of the AP decision variables; the reasoning behind this choice is explained. The decision variable φ_{ijs} of AP indicates an assignment of satellite i to orbital slot j of subconstellation s while the decision variable x_{js} of MCP indicates whether a satellite occupies orbital slot j of subconstellation s. Therefore, it follows naturally that φ_{ijs} are the elemental decision variables because we can deduce x_{js} from φ_{ijs} (see Figure 3.5). The following relationship couples these two different sets of decision variables along with Constraints (3.8b) and (3.8c):

$$x_{js} = \sum_{i \in \mathcal{I}} \varphi_{ijs}, \quad \forall j \in \mathcal{J}, \forall s \in \mathcal{S} \tag{3.9}$$

where both φ_{ijs} and x_{js} are binary variables. This coupled relationship in Eq. (3.9) enables an integrated ILP formulation that simultaneously considers both the constellation transfer problem and the constellation design problem.

3.4.3 Model Characteristics

Remark 5. The RCRP formulation possesses following characteristics:

(i) RCRP is NP-hard because of the embedded MCP structure (cf. Formulation 3). This deduction follows from the NP-hardness of MCLP [71], which has shown to be a

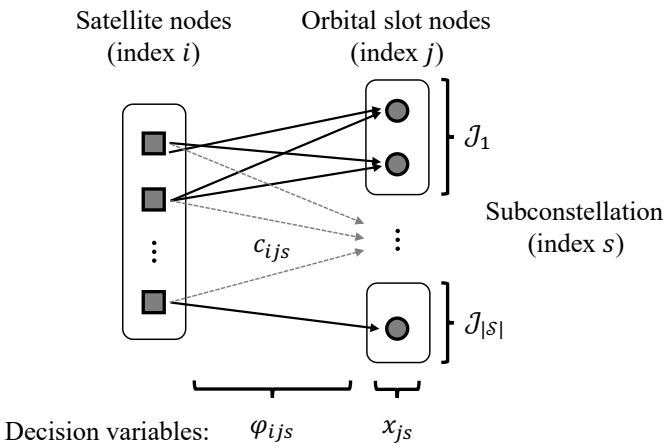


Figure 3.5: Decision variables of AP and MCP and their relationship.

particular case of MCP (see discussion in Section 3.3).

(ii) The AP structure [Constraints (3.8b),(3.8c), and (3.8e)] is preserved in RCRP with the decision variables being AP-based. In this perspective, the complicating constraints are Constraints (3.8d).

The RCRP formulation couples constellation transfer problem with the AP formulation and constellation design problem with the MCP formulation. The former exhibits a special structure that allows an efficient solution approach. The latter, on the other hand, is a combinatorial optimization problem, and consequently, the use of exact methods such as the branch-and-bound algorithm can be computationally expensive. In light of this observation, we construct a heuristic method in Section 3.5 that leverages the characteristics of the problem formalized in Remark 5.

3.5 Lagrangian Heuristic Solution Method

This section develops a heuristic solution method to RCRP, which is a bi-objective combinatorial optimization problem. To approach the bi-objective formulation, we use the ε-constraint method [73] to transform RCRP into a single-objective optimization problem, which is then solved in series by varying ε value. The transformed single-objective problem

can be solved using a commercial MILP solver, but this approach can become computationally challenging even for moderately-sized instances. Motivated by this background, we propose a computationally-efficient Lagrangian relaxation-based heuristic method that leverages the unique structure of the model.

3.5.1 ε-constraint Reformulation

The goal of RCRP, as implied by its bi-objective formulation, is to identify non-dominated solutions. To solve this problem, we reformulate RCRP as a single-objective optimization problem via the ε-constraint method by casting one of the two objective functions into a constraint with an upper (or a lower) bound ε. Solving a succession of single-objective ε-constrained problems to optimality given the sequence of ε values yields the set of non-dominated solutions, that is, the Pareto front, of the original problem. Algorithm 2 outlines the procedure based on the ε-constraint method.

Applying the ε-constraint method to RCRP, we transform the cost minimization objective function into a constraint that is bounded from above by ε [Constraint (3.10)]. Selecting an appropriate objective function for the constraint transformation is important because the choice made in this step affects the downstream algorithmic efforts; we will revisit this discussion in Section 3.5.6. In a physical sense, ε represents the maximum allowable aggregated cost of reconfiguration, hence the name *aggregated resource constraint* (ARC) for Constraint (3.10). The following is the single-objective model with the ARC:

$$\begin{aligned} \text{(RCRP-ARC)} \quad Z(\varepsilon) = \min \quad & -\sum_{p\in\mathcal{P}}\sum_{t\in\mathcal{T}} \pi_{tp} y_{tp} \\ \text{s.t.} \quad & \sum_{s\in\mathcal{S}}\sum_{j\in\mathcal{J}_s}\sum_{i\in\mathcal{I}} c_{ijs}\varphi_{ijs} \le \varepsilon \qquad (3.10) \\ & \text{Constraints (3.8b)–(3.8f)} \end{aligned}$$

where $Z(\varepsilon)$ denotes the optimal value of RCRP-ARC with parameter ε. The goal of RCRP-

ARC is to maximize the total coverage reward while being subjected to the aggregated resource constraint.

Algorithm 2: ε-constraint method

Input: $\boldsymbol{c}, \boldsymbol{\pi}, \boldsymbol{r}, \boldsymbol{v}$
Output: List of Z(ε) values
Initialize ε_0
repeat
    Z(ε) $\leftarrow$ RCRP-ARC
    $\varepsilon \leftarrow \varepsilon + \varepsilon_{\text{step}}$
until *termination flag is triggered*

3.5.2 Lagrangian Relaxation

The Lagrangian relaxation is a decomposition-based optimization technique to approach a complex problem by dualizing complicating constraints such that the remaining "relatively easy" structure is exposed and thus efficiently solved (see Ref. [74] for a general overview of the topic). Specifically, in our case, the complicating constraints can be viewed as those of MCP, Constraints (3.8d), primarily due to the intact AP structure (Remark 5) in the relaxed problem along with Constraint (3.10).

To retrieve the *Lagrangian problem* (LR) of RCRP-ARC, we dualize Constraints (3.8d):

$$
\begin{aligned}
\text{(LR)} \quad Z_{\text{D}}(\varepsilon, \boldsymbol{\lambda}) = \min \quad & -\sum_{p\in\mathcal{P}}\sum_{t\in\mathcal{T}} \pi_{tp} y_{tp} + \sum_{p\in\mathcal{P}}\sum_{t\in\mathcal{T}} \lambda_{tp}\left[r_{tp} y_{tp} - \sum_{s\in\mathcal{S}}\sum_{j\in\mathcal{J}_s}\sum_{i\in\mathcal{I}} V_{tjps}\varphi_{ijs} \right] \\
\text{s.t.} \quad & \sum_{s\in\mathcal{S}}\sum_{j\in\mathcal{J}_s} \varphi_{ijs} = 1, \quad \forall i \in \mathcal{I} \\
& \sum_{i\in\mathcal{I}} \varphi_{ijs} \leq 1, \quad \forall j \in \mathcal{J}_s, \forall s \in \mathcal{S} \\
& \sum_{s\in\mathcal{S}}\sum_{j\in\mathcal{J}_s}\sum_{i\in\mathcal{I}} c_{ijs}\varphi_{ijs} \leq \varepsilon \\
& \varphi_{ijs} \in \{0,1\}, \quad \forall i \in \mathcal{I}, \forall j \in \mathcal{J}_s, \forall s \in \mathcal{S} \\
& y_{tp} \in \{0,1\}, \quad \forall t \in \mathcal{T}, \forall p \in \mathcal{P}
\end{aligned}
$$

where $\boldsymbol{\lambda} = (\lambda_{tp} \in \mathbb{R}_{\geq 0} : t \in \mathcal{T}, p \in \mathcal{P})$ is a vector of Lagrange multipliers associated with Constraints (3.8d), and $Z_{\mathrm{D}}(\varepsilon, \boldsymbol{\lambda})$ denotes the optimal value of LR.

Remark 6. For all non-negative $\boldsymbol{\lambda}$, we have $Z_{\mathrm{D}}(\varepsilon, \boldsymbol{\lambda}) \leq Z(\varepsilon)$. It is easy to see this because for a given optimal solution $(\boldsymbol{\varphi}^*, \boldsymbol{y}^*)$ to RCRP-ARC, we observe that the following series of inequalities hold:

$$\begin{aligned} Z(\varepsilon) &\geq -\sum_{p\in\mathcal{P}}\sum_{t\in\mathcal{T}} \pi_{tp} y^*_{tp} + \sum_{p\in\mathcal{P}}\sum_{t\in\mathcal{T}} \lambda_{tp}\left[r_{tp} y^*_{tp} - \sum_{s\in\mathcal{S}}\sum_{j\in\mathcal{J}_s}\sum_{i\in\mathcal{I}} V_{tjps}\varphi^*_{ijs}\right] \\ &\geq Z_{\mathrm{D}}(\varepsilon, \boldsymbol{\lambda}) \end{aligned}$$

where the first inequality follows from adding the non-positive term to $Z(\varepsilon)$. The second inequality follows from the fact that relaxing Constraint (3.8d) potentially enlarges the feasible region, and therefore an improving solution can be found to further reduce the objective function value.

3.5.3 Lagrangian Dual Problem and Subgradient Method

Following from Remark 6, we observe that the lower bound $Z_{\mathrm{D}}(\varepsilon, \boldsymbol{\lambda})$ can be tightened up (i.e., maximized) by solving for the optimal $\boldsymbol{\lambda}^*$. Such a problem is called the *Lagrangian dual problem* (D) and is formulated as follows:

$$\text{(D)} \quad Z_{\mathrm{D}}(\varepsilon) = \max_{\boldsymbol{\lambda}} Z_{\mathrm{D}}(\varepsilon, \boldsymbol{\lambda})$$

The Lagrangian dual problem is a non-differentiable optimization problem because $Z_{\mathrm{D}}(\varepsilon, \boldsymbol{\lambda})$ is a piecewise linear concave function of $\boldsymbol{\lambda}$. To solve this problem, we use the subgradient method [75], which has shown to be an effective method for non-differentiable optimization problems [76]. The subgradient method is an iterative algorithm in the spirit of the gradient ascent method for finding the maximum solution of a continuous differentiable function. Algorithm 3 provides the pseudo-code for the subgradient optimization.

Algorithm 3: Subgradient optimization

Input: $\varepsilon, \boldsymbol{c}, \boldsymbol{\pi}, \boldsymbol{r}, \boldsymbol{v}$
Output: $Z_{\mathrm{D}}(\varepsilon), \hat{Z}(\varepsilon), (\boldsymbol{\varphi}^*, \tilde{\boldsymbol{y}}^*)$

$k \leftarrow 0$ and initialize $\boldsymbol{\lambda}^0$
repeat
  Solve LR^k: compute $Z_{\mathrm{D}}(\varepsilon, \boldsymbol{\lambda}^k)$ and obtain optimal solution $(\boldsymbol{\varphi}^k, \boldsymbol{y}^k)$ ▷ `LB; See Remark 8`
  Compute $\hat{Z}(\varepsilon)$ and $(\boldsymbol{\varphi}^*, \tilde{\boldsymbol{y}}^*)$ via local search in the neighborhood $\mathcal{N}(\boldsymbol{\varphi}^k)$ of $\boldsymbol{\varphi}^k$ ▷ `UB; See Algorithm 4`
  Compute the subgradient $\boldsymbol{g}^k$ of $Z_{\mathrm{D}}(\varepsilon, \boldsymbol{\lambda}^k)$ at $\boldsymbol{\lambda}^k$
  Compute the step size θ_k
  Update Lagrange multipliers: $\boldsymbol{\lambda}^{k+1} \leftarrow \max(\mathbf{0}, \boldsymbol{\lambda}^k + \theta_k \boldsymbol{g}^k)$
  $k \leftarrow k + 1$
until *termination flag is triggered*

The subgradient method begins with the initialization of Lagrange multipliers, $\boldsymbol{\lambda}^0$. At iteration k, and given parameters, ε, $\boldsymbol{\lambda}^k$, $\boldsymbol{c}$, $\boldsymbol{\pi}$, $\boldsymbol{r}$, and $\boldsymbol{v}$, the Lagrangian problem LR^k is solved. With the optimal solution $(\boldsymbol{\varphi}^k, \boldsymbol{y}^k)$ to LR^k, we apply a heuristic method to obtain an estimate $\hat{Z}(\varepsilon)$ of $Z(\varepsilon)$. Because we wish to retain the best estimate of the optimal value not only for the feasible primal solution but also to facilitate the convergence of the subgradient method, the best $\hat{Z}(\varepsilon)$ up to iteration k is stored in memory as an incumbent optimum.

Next, the subgradient $\boldsymbol{g}^k$ of $Z_{\mathrm{D}}(\varepsilon, \boldsymbol{\lambda}^k)$ at $\boldsymbol{\lambda}^k$ is computed:

$$\boldsymbol{g}^k = \left(r_{tp} y^k_{tp} - \sum_{s \in \mathcal{S}} \sum_{j \in \mathcal{J}_s} \sum_{i \in \mathcal{I}} V_{tjps} \varphi^k_{ijs} : t \in \mathcal{T}, p \in \mathcal{P} \right)$$

If the subgradient of $Z_{\mathrm{D}}(\varepsilon, \boldsymbol{\lambda}^k)$ at $\boldsymbol{\lambda}^k$ is $\mathbf{0}$, then $\boldsymbol{\lambda}^k$ is an optimal Lagrange multiplier vector, and the algorithm terminates. The algorithm may terminate with suboptimal Lagrangian multipliers if it triggers one or more of the following termination criteria: the maximum iteration count, the gap tolerance between $\hat{Z}(\varepsilon)$ and $Z_{\mathrm{D}}(\varepsilon, \boldsymbol{\lambda}^k)$, and the step size tolerance. With $\boldsymbol{\lambda}^0 = \mathbf{0}$, the Lagrangian relaxation bound starts with $Z_{\mathrm{D}}(\varepsilon, \boldsymbol{\lambda}^0) = -\sum_{p \in \mathcal{P}} \sum_{t \in \mathcal{T}} \pi_{tp}$ and improves as the subgradient method progresses. In case of premature termination due to reaching the maximum iteration limit, the obtained $\boldsymbol{\lambda}$ may end up suboptimal, and hence

$Z_{\mathrm{D}}(\varepsilon, \boldsymbol{\lambda}) < Z_{\mathrm{D}}(\varepsilon)$. With the knowledge of the optimal dual variables of the LP relaxation problem, one can use them for $\boldsymbol{\lambda}^0$. However, this approach is not ideal because the LP relaxation problem needs to be run beforehand, which can be computationally expensive for large instances.

Unless the termination flag is triggered, the algorithm reiterates the procedure with the new set of Lagrange multipliers. The following is the rule for updating the Lagrange multipliers:

$$\boldsymbol{\lambda}^{k+1} := \max(\mathbf{0}, \boldsymbol{\lambda}^k + \theta_k \boldsymbol{g}^k)$$

where $\max(\cdot)$ is the element-wise maximum to guarantee the non-negativity of λ_{tp}. Unless the dualized constraints are equality constraints, which can have associated multipliers unrestricted in sign, the multipliers need to be non-negative to penalize the violated constraints correctly [77].

The step size θ_k commonly used in practice is:

$$\theta_k := \frac{\hat{Z}(\varepsilon) - Z_{\mathrm{D}}(\varepsilon, \boldsymbol{\lambda}^k)}{\|\boldsymbol{g}^k\|^2} \alpha_k \tag{3.11}$$

where $\|\cdot\|$ is the Euclidean norm and α_k is a scalar satisfying $0 < \alpha_k \leq 2$. The proof of convergence of the above step size formula is referred to Ref. [76]. Recommended by Fisher [74], the starting value of α_k is set to $\alpha_0 = 2$ and is halved if $Z_{\mathrm{D}}(\varepsilon, \boldsymbol{\lambda})$ fails to increase in a number of iterations. While there are different types of θ_k proposed in literature, the step size formula in Eq. (3.11) has performed particularly well in our problem settings.

The subgradient method suffers from several drawbacks such as the zigzagging phenomenon and slow convergence to the optimal multipliers $\boldsymbol{\lambda}^*$. Many studies have proposed variants of the subgradient method such as the surrogate Lagrangian relaxation method and the bundle method to alleviate such issues. Interested readers are referred to Ref. [78] for additional materials on methods for non-differentiable problems.

There exist two computational bottlenecks in this algorithm. One at computing the

lower bound $Z_{\mathrm{D}}(\varepsilon, \boldsymbol{\lambda}^k)$ and another at computing the upper bound $\hat{Z}(\varepsilon)$, both of which occur at every iteration. To solve each iteration of the subgradient method efficiently, we provide efficient ways to compute $Z_{\mathrm{D}}(\varepsilon, \boldsymbol{\lambda}^k)$ and $\hat{Z}(\varepsilon)$.

3.5.4 Lower Bound: Lagrangian Problem Decomposition

At each iteration of the subgradient optimization, $Z_{\mathrm{D}}(\varepsilon, \boldsymbol{\lambda}^k)$ is computed. By relaxing the complicating constraints [Constraints (3.8d)], which are also the linking constraints, we observe that Problem (LR) can be decomposed into two subproblems based on the variable type, $\boldsymbol{\varphi}$ and $\boldsymbol{y}$.

Remark 7. Problem (LR) can be decomposed into two subproblems: (i) an assignment problem with a side constraint and (ii) an unconstrained binary integer linear program.

$$
\begin{aligned}
\text{(LR1)} \quad Z_{\mathrm{D1}}(\varepsilon, \boldsymbol{\lambda}) = \min \quad & -\sum_{p\in\mathcal{P}}\sum_{t\in\mathcal{T}} \lambda_{tp}\Bigg[\sum_{s\in\mathcal{S}}\sum_{j\in\mathcal{J}_s}\sum_{i\in\mathcal{I}} V_{tjps}\varphi_{ijs}\Bigg] \\
\text{s.t.} \quad & \sum_{s\in\mathcal{S}}\sum_{j\in\mathcal{J}_s} \varphi_{ijs} = 1, && \forall i \in \mathcal{I} \\
& \sum_{i\in\mathcal{I}} \varphi_{ijs} \le 1, && \forall j \in \mathcal{J}_s, \forall s \in \mathcal{S} \\
& \sum_{s\in\mathcal{S}}\sum_{j\in\mathcal{J}_s}\sum_{i\in\mathcal{I}} c_{ijs}\varphi_{ijs} \le \varepsilon \\
& \varphi_{ijs} \in \{0,1\}, && \forall i \in \mathcal{I}, \forall j \in \mathcal{J}_s, \forall s \in \mathcal{S}
\end{aligned}
$$

$$
\begin{aligned}
\text{(LR2)} \quad Z_{\mathrm{D2}}(\boldsymbol{\lambda}) = \min \quad & \sum_{p\in\mathcal{P}}\sum_{t\in\mathcal{T}} (\lambda_{tp} r_{tp} - \pi_{tp}) y_{tp} \\
\text{s.t.} \quad & y_{tp} \in \{0,1\}, && \forall t \in \mathcal{T}, \forall p \in \mathcal{P}
\end{aligned}
$$

For convenience, we denote $\Phi := \{\boldsymbol{\varphi} \in \{0,1\}^{|\mathcal{I}||\mathcal{J}|} : \text{Constraints (3.8b), (3.8c), (3.10)}\}$ the set of assignments $\boldsymbol{\varphi}$ satisfying the AP and ARC constraints.

Remark 8. Problem (LR) is characterized as follows:

(i) Let ε_{r} denote the critical value of ε at which Constraint (3.10) becomes *redundant* in Co(Φ), the convex hull of the feasible region of Problem (LR1). From inspection, it is evident that no total cost of assignments exceeds the value of

$$\varepsilon_{\mathrm{r}} := \sum_{i \in \mathcal{I}} \max_{j \in \mathcal{J}_s, s \in \mathcal{S}} c_{ijs}$$

If $\varepsilon \geq \varepsilon_{\mathrm{r}}$, then Constraint (3.10) is redundant and can be removed while leaving the convex hull unchanged; this reveals the intact AP structure [Constraints (3.8b),(3.8c), and (3.8e)], which possesses the integrality property (cf. Formulation 2). In this case, Problem (LR1) can be efficiently solved using a specialized AP algorithm or via linear programming. Otherwise, Problem (LR1) does not possess the integrality property, and therefore the problem cannot be solved by means of LP. Based on our experience, the problem can be considered a relatively easy ILP and thus be expected to solve in a reasonable amount of time in many instances, albeit there are instances where guaranteeing the solution optimality takes a long time.

(ii) Problem (LR2) can be solved trivially. For fixed multipliers, the optimal solution is generated by inspecting the coefficients of the objective function:

$$y^*_{tp} = \begin{cases} 1, & \text{if } (\lambda_{tp} r_{tp} - \pi_{tp}) < 0 \\ 0, & \text{otherwise} \end{cases}$$

(iii) The optimal value of Problem (LR) is the sum of the optimal values of its subproblems: $Z_{\mathrm{D}}(\varepsilon, \boldsymbol{\lambda}) = Z_{\mathrm{D1}}(\varepsilon, \boldsymbol{\lambda}) + Z_{\mathrm{D2}}(\boldsymbol{\lambda})$.

Using the argument of LP duality [79], one can deduce that the Lagrangian relaxation bound is at least as good as the LP relaxation bound: $Z_{\mathrm{LP}}(\varepsilon) \leq Z_{\mathrm{D}}(\varepsilon)$. Based on observations made in Remark 8, we provide a discussion on the tightness of these two bounds and a

sufficient condition at which they are equal.

Remark 9. The relative strength of $Z_{\mathrm{D}}(\varepsilon)$ with respect to $Z_{\mathrm{LP}}(\varepsilon)$ depends on the redundancy of Constraint (3.10) in Problem LR1 discussed in Remark 8 (i).

(i) If Constraint (3.10) is not redundant, the Lagrangian relaxation may provide a tighter bound than the LP relaxation bound: $Z_{\mathrm{LP}}(\varepsilon) \leq Z_{\mathrm{D}}(\varepsilon)$. The value of ε dictates the tightness of these bounds.

(ii) If Constraint (3.10) is redundant, Problem (LR) possesses the integrality property, which is a sufficient condition for $Z_{\mathrm{LP}}(\varepsilon) = Z_{\mathrm{D}}(\varepsilon)$ [79].

Reiterating Remark 9 (ii), Constraint (3.10) being redundant is only a sufficient condition for the equal bounds, and therefore it does not necessarily imply $Z_{\mathrm{LP}}(\varepsilon) < Z_{\mathrm{D}}(\varepsilon)$ for all $\varepsilon < \varepsilon_{\mathrm{r}}$. In some instances, we observe that there exists $\varepsilon_{\mathrm{c}} < \varepsilon_{\mathrm{r}}$ such that $Z_{\mathrm{LP}}(\varepsilon) < Z_{\mathrm{D}}(\varepsilon)$ for $\varepsilon < \varepsilon_{\mathrm{c}}$ (and $Z_{\mathrm{LP}}(\varepsilon) = Z_{\mathrm{D}}(\varepsilon)$ for $\varepsilon \geq \varepsilon_{\mathrm{c}}$). Note that if $Z_{\mathrm{LP}} = Z_{\mathrm{D}}$, then the optimal Lagrange multipliers are equal to the dual variables associated with Constraints (3.8d) of the LP relaxation of RCRP-ARC.

Revisit the earlier discussion on the LP relaxation bound for MCP discussed in Section 3.3. There is a special case, that is, $\xi_t = \xi, \forall t \in \mathcal{T}$ and $\boldsymbol{V}$ being a circulant matrix, for which the value of $\hat{Z}_{\mathrm{LP}}$ is independent of optimal LP solution $\boldsymbol{x}^*$. This analysis is readily extensible to the RCRP-ARC formulation. Consequently, for $\varepsilon \geq \varepsilon_{\mathrm{r}}$, $\xi_{tp} = \xi_p, \forall t \in \mathcal{T}$, and all relevant $\boldsymbol{V}$ matrices being circulant, the approximated LP relaxation bound for RCRP-ARC is simply [cf. Eq. (3.7)]:

$$\hat{Z}_{\mathrm{LP}} \coloneqq \max\Bigg(- |\mathcal{I}| \sum_{p \in \mathcal{P}} \sum_{t \in \mathcal{T}} \xi_{tp} v_{tp}, - \sum_{p \in \mathcal{P}} \sum_{t \in \mathcal{T}} \pi_{tp} \Bigg)$$

where $\xi_{tp} \coloneqq \pi_{tp} / r_{tp}$.

The value of $\hat{Z}_{\mathrm{LP}}$ provides the conservative lower bound for both Z_{LP} and Z_{D} that requires no knowledge of the optimal LP solution $\boldsymbol{\varphi}^*$. As will be discussed later with the

computational experiments in Section 3.7, we can actually see that $\hat{Z}_{\mathrm{LP}}$ outperforms Z_{D} for suboptimal Lagrange multipliers. However, its usefulness is limited to instances triggering the first argument of the max function.

3.5.5 Upper Bound: Lagrangian Heuristic via Combinatorial Neighborhood Local Search

To evaluate the duality gap over the convergence (e.g., to be used for the definition of the step size), we need a method to find a feasible solution to the primal problem (RCRP-ARC) at each iteration step k of the subgradient method. While it is theoretically possible for a solution to LR^k to be discovered to be feasible to the primal problem in the course of solving the Lagrangian dual problem, this happens rarely [74]; instead, we attempt to convert the solution to LR^k into a feasible solution to the primal problem. To this end, we note that the reason for solution infeasibility is due to the inconsistency between weakly coupled $\boldsymbol{\varphi}^k$ and $\boldsymbol{y}^k$ solutions; the coupling comes from the fact that Problems $\mathrm{LR1}^k$ and $\mathrm{LR2}^k$ are linked only through $\boldsymbol{\lambda}^k$ in their objective functions. Intuitively speaking, the coverage state of the obtained constellation configuration, which is computed from the assignments using Eq. (3.12), does not match with the obtained coverage state. The key idea to find a feasible solution is to drive the solution obtained at each iteration of the subgradient method to feasibility by employing a heuristic approach. Such an approach that exploits a solution produced in the course of solving the Lagrangian dual problem is called the *Lagrangian heuristic* in literature.

A simple and straightforward way is to accept the subgradient assignment $\boldsymbol{\varphi}^k$ as the valid primal solution and obtain the conforming coverage state $\tilde{\boldsymbol{y}}^k(\boldsymbol{\varphi}^k)$ (note that we distinguish it from $\boldsymbol{y}^k$, without a tilde), which is simply the coverage state of the constellation configuration obtained from the set of assignments $\boldsymbol{\varphi}^k$:

$$\tilde{y}^k_{tp}(\varphi^k_{ijs}) = \begin{cases} 1, & \text{if } \sum_{s\in\mathcal{S}} \sum_{j\in\mathcal{J}_s} \sum_{i\in I} V_{tjps}\varphi^k_{ijs} \geq r_{tp} \\ 0, & \text{otherwise} \end{cases} \tag{3.12}$$

This approach always yields feasible solutions to the primal problem because it circumvents the inconsistency between φ^k and $\boldsymbol{y}^k$. The other way around, obtaining $\tilde{\varphi}^k(\boldsymbol{y}^k)$ from given $\boldsymbol{y}^k$, is infeasible in most cases or requires a combinatorial optimization approach in otherwise rarely feasible cases.

A more sophisticated approach is to employ a local search in the neighborhood of φ^k to improve the quality of the initial primal solution at the cost of additional computation time. In this chapter, we propose the following definition of the *1-exchange neighborhood*:

$$\mathcal{N}(\varphi) \coloneqq \{\varphi' \in \Phi : \varphi' \text{ obtained from } \varphi \text{ by } \textit{exchanging at most one (satellite, orbital slot) pair}\} \quad (3.13)$$

Figure 3.6 illustrates the 1-exchange operation. Two sets of assignments φ and φ' are *neighbors* because at most a single (satellite, orbital slot) assignment pair is different (the dashed line in the figure). Orbital slots that are occupied by other satellites are not part of the neighborhood because it violates the definition of the feasible set Φ. Not all unfilled orbital slots are valid candidates because all candidate orbital slots must conform with Constraint (3.10).

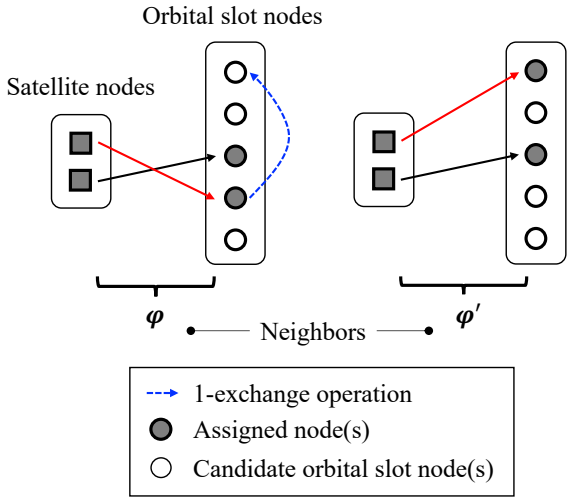


Figure 3.6: Illustration of the 1-exchange operation.

The 1-exchange neighborhood local search algorithm is presented in Algorithm 4. Given φ, the 1-exchange neighborhood $\mathcal{N}(\varphi)$ is defined as in Eq. (3.13). The best candidate solution φ^* is found from an exhaustive evaluation of all candidate solutions in $\mathcal{N}(\varphi)$. Here, the figure of merit is $\hat{Z}(\varepsilon)$. If the best candidate solution φ^* outperforms the incumbent solution, then φ^* is accepted as the new incumbent solution, and $\hat{Z}(\varepsilon)$ and the neighborhood are updated accordingly. The search process reiterates with the new neighborhood. Otherwise, the local search halts and the local optimum is obtained. The algorithm returns the local optimal solution φ^*, its conforming $\tilde{\boldsymbol{y}}^*(\varphi^*)$ [Eq. (3.12)], and the corresponding upper bound value $\hat{Z}(\varepsilon)$.

Algorithm 4: 1-exchange neighborhood local search (full-scale)

Input: $\varepsilon, \boldsymbol{c}, \boldsymbol{\pi}, \boldsymbol{r}, \boldsymbol{v}, \boldsymbol{\lambda}, \varphi$
Output: $\hat{Z}(\varepsilon), (\varphi^*, \tilde{\boldsymbol{y}}^*(\varphi^*))$
1 Initialize $\mathcal{N}(\varphi)$
2 **repeat**
3 | Compute $\hat{Z}(\varepsilon)$ and φ^* from $\mathcal{N}(\varphi)$
4 | Update $\varphi \leftarrow \varphi^*$
5 | Update the neighborhood $\mathcal{N}(\varphi)$
6 **until** *termination flag is triggered*
7 Compute $\tilde{\boldsymbol{y}}^*(\varphi^*)$ from φ^*

The trade-off between the solution quality and the computational effort exists in Line 3 of Algorithm 4 at deciding the local optimum of the current neighborhood. The brute-force evaluation of all candidate solutions can be costly for large instances, hence it is practical to search only within the subset of a neighborhood $\mathcal{N}' \subseteq \mathcal{N}$, which can be randomly generated or via a pre-defined rule. Moreover, the first-come-first-served scheme can be applied to select the first local solution that improves the incumbent solution, which can reduce the dimension of the search space. Noting that the radius of a neighborhood can play a significant role in the trade-off, leveraging more general κ-exchange neighborhood local search ($\kappa > 1$) could improve the quality of the heuristic solution for large instances at the expense of an increased computational effort.

3.5.6 Selecting between Competing Relaxations

There exist different types of Lagrangian relaxations for RCRP, and a choice of ε-constraint transformation influences the complexity of the downstream algorithmic efforts and the mathematical properties. At a glance, one may observe that the coverage reward maximization objective function can recast as an ε-constraint. Similar to the one proposed in this chapter, the Lagrangian relaxation problem would be separable into two subproblems based on the type of variables. In such a case, the $\boldsymbol{\varphi}$ subproblem can be solved as an LP, and the $\boldsymbol{y}$ subproblem would be a relatively easy constrained ILP. Therefore, the lower bound calculation would still be computationally efficient. However, the main difference lies in the computation of $\hat{Z}(\varepsilon)$: unlike the one discussed earlier, $\tilde{\boldsymbol{y}}^k(\boldsymbol{\varphi}^k)$ computed from $\boldsymbol{\varphi}^k$ would not necessarily satisfy the ε-constraint. Hence, additional considerations must come into play in obtaining the feasible primal solution. One viable approach is to solve the reduced formulation of RCRP, which fixes and parameterizes a subset of assignments from $\boldsymbol{\varphi}^k$ while optimizing the complement set. However, this approach becomes computationally expensive for instances with high ε values. This approach was explored in our preliminary work [80].

One could attempt to relax an alternative set of constraints that may yield a tighter Lagrangian relaxation bound than the one proposed in this subsection. However, Lagrangian relaxation problems with Constraints (3.8d) retained may be unsuitable for embedding into an algorithm of iterative nature due to computational complexity. The rationale for the relaxation of Constraints (3.8d) can also be found in a study by Galvao and ReVelle [81], which reported a successful application of the Lagrangian relaxation of the linking constraints for MCLP; in their problem context, the Lagrangian relaxation problem possesses the integrality property.

3.6 Extension: RCRP with Individual Resource Constraints

Various extensions can be made to the proposed formulation and method. One example that is practically important is shown here. Constraint (3.10) in the RCRP-ARC formulation limits the aggregated resource consumed by all system satellites. Along a similar line, but instead, we can formulate a variant of RCRP to enforce the maximum individual resource consumption on each satellite. Such modeling possesses significant practical implications because not all satellites have identical resource states prior to a reconfiguration. As an example, one can envisage a realistic scenario of bringing a group of satellites with different fuel states together to form a federation for a new Earth observation mission [82].

We formulate the RCRP with *individual resource constraints* (RCRP-IRC) as follows:

$$\begin{aligned} \text{(RCRP-IRC)} \quad \min \quad & -\sum_{p\in\mathcal{P}}\sum_{t\in\mathcal{T}} \pi_{tp} y_{tp} \\ \text{s.t.} \quad & \sum_{s\in\mathcal{S}}\sum_{j\in\mathcal{J}_s} c_{ijs}\varphi_{ijs} \le \varepsilon_i, \quad \forall i \in \mathcal{I}' \qquad (3.14) \\ & \text{Constraints (3.8b)–(3.8f)} \end{aligned}$$

where $\mathcal{I}' \subseteq \mathcal{I}$ is the subset of satellites with IRC; Constraints (3.14) define the reachable domain of orbital slots by satellite i given the maximum allowance ε_i.

The proposed solution procedure based on the Lagrangian relaxation method for RCRP-ARC is readily applicable to RCRP-IRC. This is true because the Lagrangian problem for RCRP-IRC is also separable into two subproblems based on the variable type, and the primal heuristic can be readily applied with the modified definition of Φ. Note that the analyses made in Remarks 8 and 9 can be extended to RCRP-IRC with the appropriate values of $\varepsilon_{\mathrm{r},i}$.

3.7 Computational Experiments

We perform computational experiments to evaluate the performance of the proposed Lagrangian heuristic method. In particular, we are interested in analyzing the solution quality and the computational efficiency of the Lagrangian heuristic compared to the results obtained by a commercial off-the-shelf MIP solver. We first perform the design of experiments in Section 3.7.1 and then compare the results obtained by the Lagrangian heuristic and commercial software package in Section 3.7.2. The primary computational experiments are performed using RCRP-ARC for RGT orbits. In Section 3.7.4, we provide an illustrative example to demonstrate the versatility of the proposed framework by extending it to non-RGT orbits and RCRP-IRC.

3.7.1 Test Instances

We generate test instances by varying $|\mathcal{I}|$, $|\mathcal{J}|$, $|\mathcal{T}|$, and $|\mathcal{P}|$. For each size dimension, we take the following value: $|\mathcal{I}| \in \{10, 20\}$, $|\mathcal{J}| \in \{500, 1000, 2000\}$, and $|\mathcal{P}| \in \{10, 20, 30\}$. Without loss of generality we let $|\mathcal{T}| = |\mathcal{J}|$. Table 3.1 shows the sizes of the randomly generated 18 test instances for RCRP; the number of constraints exclude the domain definitions of decision variables. The size of the instance pool varies from the smallest containing 10,000 decision variables and 5,511 constraints to the largest containing 100,000 decision variables and 62,021 constraints. Note that there are $|\mathcal{I}||\mathcal{J}|$ number of $\boldsymbol{\varphi}$ decision variables and $|\mathcal{T}||\mathcal{P}|$ number of $\boldsymbol{y}$ decision variables for both RCRP-ARC and RCRP-IRC. There are $|\mathcal{I}| + |\mathcal{J}| + |\mathcal{T}||\mathcal{P}| + 1$ constraints for RCRP-ARC and $|\mathcal{I}| + |\mathcal{J}| + |\mathcal{T}||\mathcal{P}| + |\mathcal{I}'|$ constraints for RCRP-IRC excluding the domain definitions of decision variables. For each RCRP test instance, there are 10 RCRP-ARC sub-instances with varying ε. Without loss of generality, we set $\varepsilon_{\max} = \max c_{ijs}$ and create a sequence of 10 steps in $[0, \varepsilon_{\max}]$. Consequently, we test 18 instances for RCRP, which is equivalent to testing 180 instances of RCRP-ARC.

For each RCRP instance, we specify different parameter values. Our goal is to capture

Table 3.1: Test instances for RCRP-ARC and their sizes.

Instance	$\|I\|$	$\|J\|, \|T\|$	$\|P\|$	Size of RCRP-ARC			
				φ variables	$\boldsymbol{y}$ variables	Total variables	Total constraints
1	10	500	10	5,000	5,000	10,000	5,511
2	20	500	10	10,000	5,000	15,000	5,521
3	10	500	20	5,000	10,000	15,000	10,511
4	20	500	20	10,000	10,000	20,000	10,521
5	10	1,000	10	10,000	10,000	20,000	11,011
6	20	1,000	10	20,000	10,000	30,000	11,021
7	10	500	30	5,000	15,000	20,000	15,511
8	20	500	30	10,000	15,000	25,000	15,521
9	10	1,000	20	10,000	20,000	30,000	21,011
10	20	1,000	20	20,000	20,000	40,000	21,021
11	10	2,000	10	20,000	20,000	40,000	22,011
12	20	2,000	10	40,000	20,000	60,000	22,021
13	10	1,000	30	10,000	30,000	40,000	31,011
14	20	1,000	30	20,000	30,000	50,000	31,021
15	10	2,000	20	20,000	40,000	60,000	42,011
16	20	2,000	20	40,000	40,000	80,000	42,021
17	10	2,000	30	20,000	60,000	80,000	62,011
18	20	2,000	30	40,000	60,000	100,000	62,021

a wide spectrum of orbital characteristics of orbital slot nodes and $\vartheta_{\min}$. We randomly generate 18 parameter sets from the parameter space $\{N_{\mathrm{P}} \in \mathbb{Z}_{\geq 0} : 30 \leq N_{\mathrm{P}} \leq 45\} \times \{inc \in \mathbb{R}_{\geq 0} : 0^\circ \leq inc \leq 120^\circ\} \times \{\vartheta_{\min} \in \mathbb{R}_{\geq 0} : 5^\circ \leq \vartheta_{\min} \leq 20^\circ\}$. Along with the fixed parameters shown in Table 3.2, we can characterize the orbital slot nodes as circular, prograde/retrograde low Earth orbits (specifically, the altitude ranges between 478.86 km and 2729.95 km). The sensor field-of-views range from medium to wide. In addition, we randomly generate the initial positions of $|\mathcal{I}|$ satellites in $\mathcal{J}$ from a discrete uniform distribution between 0 and $|\mathcal{J}| - 1$. The target points are randomly distributed in a region latitudinally bounded by the inclination of a given test instance; the longitude and latitude each take a value from a uniform distribution.

We compare the results of the Lagrangian heuristic and the commercial MIP solver, Gurobi 9.1.1. All computational experiments are coded in MATLAB and performed on a platform with Intel Core i7-9700 3.00 GHz CPU processor (8 cores and 8 threads) and 32 GB memory. In all cases, we let the Gurobi optimizer to utilize all 8 cores. The default

Table 3.2: Fixed parameters.

Parameter	Generation scheme
Orbital characteristics of $\mathcal{J}$	$N_D = 3$, $e = 0$ (circular orbits), $\Omega = 0°$, $u = 0°$
Coverage threshold	$r_{tp} = 1$ for all t and p
Cost matrix	Combined plane change and the Hohmann transfer maneuvers and phasing maneuvers [68]

Gurobi parameters are used except for the duality gap tolerance of 0.5 % (for both the baseline Gurobi case and (LR1)) and the runtime limit of 3600 s. For the Lagrangian heuristic, we limit the 1-exchange neighborhood local search with the size $|\mathcal{N}'| \leq 10|\mathcal{I}|$ for the primal heuristic and the Gurobi optimizer for solving Problem (LR1). In generic terms lower bound (LB) and upper bound (UB), we define the duality gap as DG = |LB − UB|/|UB|. To assess the quality of $\hat{Z}$ obtained by the Lagrangian heuristic relative to Z_G obtained by the Gurobi optimizer, we define the *relative performance metric*, RP = $(\hat{Z} - Z_G)/Z_G$, unrestricted in sign. If RP > 0, the optimum obtained by the Lagrangian heuristic outperforms that of the Gurobi optimizer. If RP < 0, the optimum obtained by the Gurobi optimizer outperforms that of the Lagrangian heuristic. If RP = 0, the obtained optimums of both methods are the same.

3.7.2 Computational Experiment Results

Out of 180 RCRP-ARC test instances, we present detailed analyses for 36 instances. Table 3.3 reports the computational results for test instances with $\varepsilon/\varepsilon_{max} = 0.3$. For 9 "small" instances, the baseline Gurobi optimizer successfully identified optimal solutions, or those within the specified duality gap tolerance of 0.5 %, within the specified time limit of 3600 s. However, as the size of instances grows, we start to observe the Gurobi optimizer triggering the time limit. Particularly, for instances 17 and 18, we see a significant duality gap of 11.03 % and 70.15 %, respectively. Examining the results of the Lagrangian heuristic, we observe that all 18 instances are solved in less than 462.24 s. Comparing the feasible primal solutions to RCRP-ARC, there are 10 instances in which Gurobi solutions perform

better than the Lagrangian heuristic solutions. However, the differences are at most 1.77 %. The Lagrangian heuristic outperforms the Gurobi optimizer for 6 instances with the largest recorded margin of 26.21 % (instance 18) and obtains optimal solutions for 2 instances.

Table 3.4 shows the results for $\varepsilon/\varepsilon_{\max} = 0.8$ cases. All parameters are the same as those we have shown previously in Table 3.3 except the ε value; an increase in the value of $\varepsilon/\varepsilon_{\max}$ leads to the enlargement of the feasible solution set. Out of 18 instances, the Gurobi optimizer solves only one instance (instance 5) to the optimality within the time limit and one instance (instance 13) to the tolerance-optimality by the time limit. The Lagrangian heuristic solves all instances with the maximum runtime of 810.97 s. The duality gaps obtained by the Lagrangian heuristic are comparably larger than those with the lower ε because Z_{D} converges to Z_{LP}. However, it is important to note that the Lagrangian relaxation bound is theoretically no worse than the LP relaxation bound (assuming converged multipliers), which in turn hints that the integrality gap of the problem is significant. For 12 out of 18 instances, the primal solutions obtained by the Lagrangian heuristic outperform those of the Gurobi optimizer. The out-performance of the Lagrangian heuristic over the Gurobi optimizer is noticeably significant for instances 14–18; the relative performance metric ranges from 11.68 % to 25.84 %. The under-performance of the Lagrangian heuristic is also observable with the relative performance metric ranging up to 1.36 %.

Table 3.3: Computational results for RCRP-ARC test instances with $\varepsilon/\varepsilon_{\max} = 0.3$.

Instance	Gurobi (8-core)				Lagrangian heuristic						RP[c], %
	LB	UB (Z_G)	Runtime[a], s	DG, %	LB (Z_D)	Runtime, s	UB ($\hat{Z}$)	Runtime, s	Total runtime[b], s	DG, %	
1	-2,701	-2,701	58.42	0	-2,790.36	3.81	-2,701	3.54	23.21	3.31	0
2	-2,860	-2,855	20.04	0.18	-2,908.58	25.06	-2,839	10.98	57.29	2.45	-0.56
3	-8,273	-8,246	626.01	0.33	-8,633.18	6.98	-8,229	18.64	45.92	4.91	-0.21
4	-7,759	-7,721	1,765.20	0.49	-8,026.34	24.36	-7,658	25.27	71.21	4.81	-0.82
5	-1,910	-1,910	7.61	0	-1,947.42	25.92	-1,910	20.49	75.00	1.96	0
6	-5,926	-5,898	439.14	0.47	-6,139.74	55.39	-5,819	20.04	97.56	5.51	-1.34
7	-10,414	-10,372	1,039.13	0.40	-10,952.97	6.82	-10,357	24.57	51.89	5.75	-0.14
8	-13,412	-13,350	1,060.46	0.46	-13,559.13	159.60	-13,291	50.35	245.75	2.02	-0.44
9	-18,530	-17,704	-	4.67	-18,565.75	22.50	-17,976	98.79	155.67	3.28	1.54
10	-16,764	-16,681	2,842.39	0.50	-17,047.76	54.95	-16,385	31.50	112.65	4.04	-1.77
11	-15,549	-15,456	-	0.60	-15,719.22	48.23	-15,436	70.60	149.94	1.83	-0.13
12	-16,788	-15,749	-	6.60	-16,838.56	66.40	-16,235	98.89	201.11	3.72	3.09
13	-13,570	-13,376	-	1.45	-13,641.58	14.10	-13,328	51.32	87.62	2.35	-0.36
14	-27,064	-25,889	-	4.54	-27,212.63	40.86	-25,846	81.93	162.77	5.29	-0.17
15	-21,399	-20,643	-	3.66	-21,523.70	25.99	-20,801	30.80	88.18	3.47	0.77
16	-32,524	-29,872	-	8.88	-32,602.67	68.43	-30,824	336.10	462.24	5.77	3.19
17	-47,164	-42,480	-	11.03	-47,302.04	40.05	-46,219	148.74	242.97	2.34	8.80
18	-60,000	-35,263	-	70.15	-46,708.36	37.08	-44,504	74.25	162.22	4.95	26.21

[a] Hyphen (-) indicates the trigger of the time limit of 3600 s.
[b] Includes runtimes for LB, UB, and the subgradient method intermediate steps.
[c] Positive RP indicates the outperformance of the Lagrangian heuristic.

Table 3.4: Computational results for RCRP-ARC test instances with $\varepsilon/\varepsilon_{\max} = 0.8$.

Instance	Gurobi (8-core)				Lagrangian heuristic						RP[c], %
	LB	UB (Z_G)	Runtime[a], s	DG, %	LB (Z_D)	Runtime, s	UB ($\hat{Z}$)	Runtime, s	Total runtime[b], s	DG, %	
1	-2,800	-2,750	-	1.82	-2,794.30	13.01	-2,725	12.24	39.15	2.54	-0.91
2	-3,002	-2,949	-	1.80	-3,036.67	36.14	-2,915	37.48	97.52	4.17	-1.15
3	-8,998	-8,404	-	7.07	-9,079.56	23.18	-8,368	43.12	85.05	8.50	-0.43
4	-8,539	-8,008	-	6.63	-8,575.73	30.51	-7,964	110.34	164.22	7.68	-0.55
5	-1,910	-1,910	24.22	0	-1,917.86	32.89	-1,910	22.45	74.05	0.41	0
6	-7,105	-6,324	-	12.35	-7,139.11	72.30	-6,332	65.31	163.22	12.75	0.13
7	-11,308	-10,936	-	3.40	-11,336.61	10.68	-10,787	39.28	63.85	5.10	-1.36
8	-14,595	-13,888	-	5.09	-14,853.03	39.79	-13,937	120.63	184.69	6.57	0.35
9	-19,706	-17,765	-	10.93	-20,000.00	12.25	-18,206	52.93	74.87	9.85	2.48
10	-19,561	-16,667	-	17.36	-19,855.75	42.07	-17,499	105.23	167.91	13.47	4.99
11	-17,067	-16,007	-	6.62	-17,154.73	52.43	-16,241	88.30	162.16	5.63	1.46
12	-17,722	-15,963	-	11.02	-17,804.39	138.79	-16,819	274.64	457.87	5.86	5.36
13	-13,600	-13,542	-	0.43	-13,651.54	24.38	-13,490	87.55	133.38	1.20	-0.38
14	-30,000	-21,791	-	37.67	-29,339.01	65.11	-27,033	316.57	427.93	8.53	24.06
15	-40,000	-17,032	-	134.85	-21,685.08	56.67	-20,908	60.93	157.88	3.72	22.76
16	-40,000	-28,719	-	39.28	-34,860.21	113.75	-32,073	643.39	810.97	8.69	11.68
17	-60,000	-42,761	-	40.31	-53,278.35	72.80	-48,932	399.13	550.60	8.88	14.43
18	-60,000	-36,585	-	64.00	-49,433.34	73.83	-46,037	232.74	366.34	7.38	25.84

[a] Hyphen (-) indicates the trigger of the time limit of 3600 s.
[b] Includes runtimes for LB, UB, and the subgradient method intermediate steps.
[c] Positive RP indicates the out-performance of the Lagrangian heuristic.

We show the results of all 180 test instances graphically. Figures 3.7 and 3.8 visualize the computational results. In these figures, all metrics, $\hat{Z}_{\mathrm{LP}}$, Z_{D}, Z_{G}, and $\hat{Z}$, are normalized and flipped in sign for convenience of physical interpretation of the results. Note that, to generate the true Pareto front of a given RCRP instance, all associated RCRP-ARC instances need to be solved to optimality.[2] Without the optimality certificate (which are typically proven by the duality gap), the results, Z_{D} and Z_{G}, in Figs. 3.7 and 3.8 are deemed the approximations of Pareto fronts; the dominated solutions are still included for completeness. Figure 3.8 corroborates the out-performance of the Lagrangian heuristic for large instances. For instances 1, 5, and 13, we observe that $\hat{Z}_{\mathrm{LP}}$ do a good job in certifying that Z_{D} is either converged or not optimal, albeit its usefulness seems to be limited. Figures 3.9 and 3.10 compare the computation runtime between the 8-core Gurobi optimizer and the Lagrangian heuristic with no parallel computing implementation (except that we solve (LR1) using the 8-core Gurobi optimizer). For almost all cases, we observe that the Gurobi optimizer reached the time limit of 3600 s. A notable case is instance 5. In this case, we see that the initial solution is near-optimal and that no significant maneuver is needed to maximize the total coverage reward.

3.7.3 Post-Lagrangian Heuristic Refinement

In this section, we examine a post-processing heuristic to refine the obtained (final) Lagrangian heuristic solution with the goal of further improving the solution quality. The idea to be explored is called the post-Lagrangian heuristic refinement procedure, which performs the *full-scale* 1-exchange neighborhood local search with the final Lagrangian heuristic solution. Note that the 1-exchange neighborhood implemented within the subgradient method is limited to a small size (e.g., $|\mathcal{N}'| \leq 10|\mathcal{I}|$ in Section 3.7.1), instead of a full scale, to facilitate the overall computational process of an iterative algorithm.

To evaluate the performance of the post-Lagrangian heuristic refinement procedure, we

[2] Strictly speaking, the Pareto front in the discrete-time domain is also the approximation of the true Pareto front in the continuous-time domain.

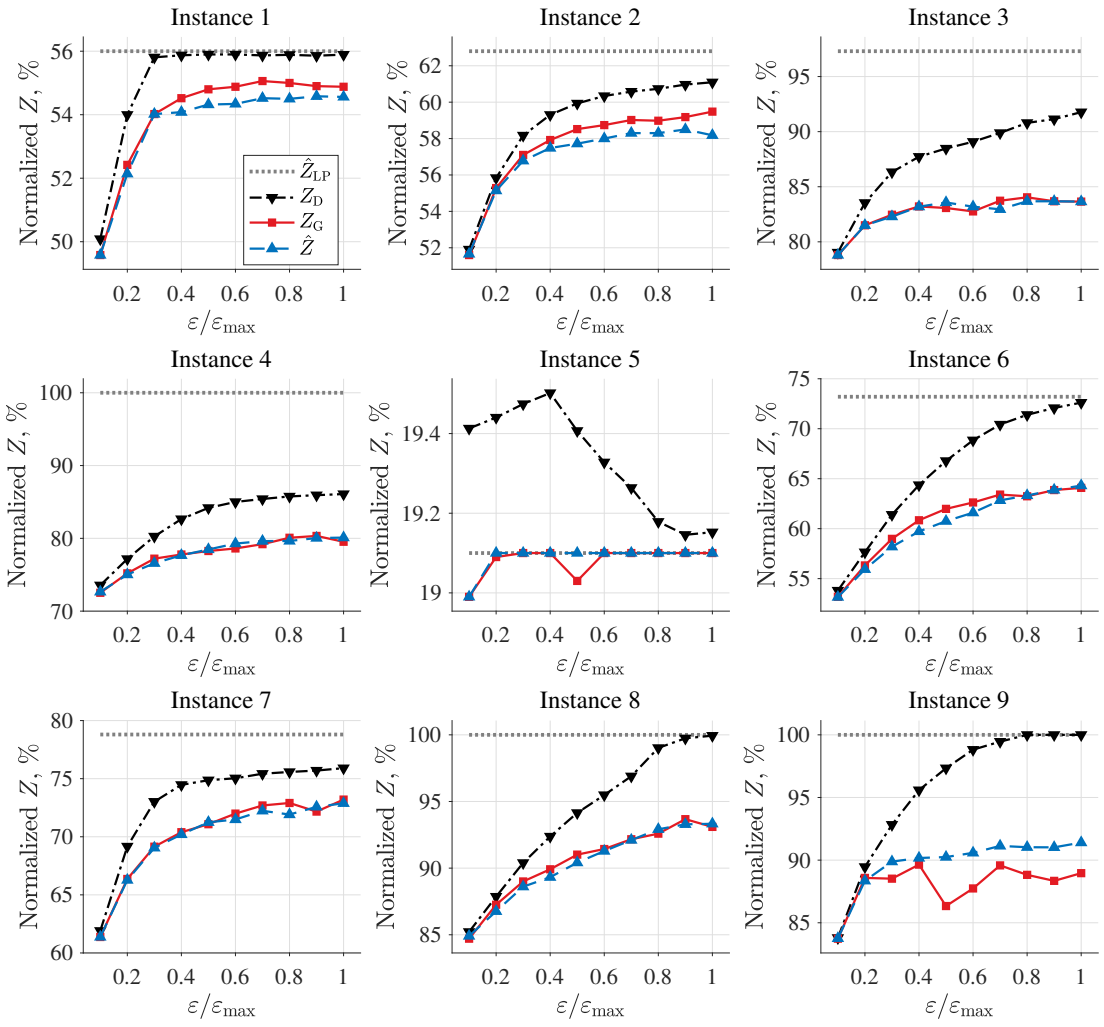


Figure 3.7: Computational results for instances 1–9. Note that all metrics are normalized and flipped in sign.

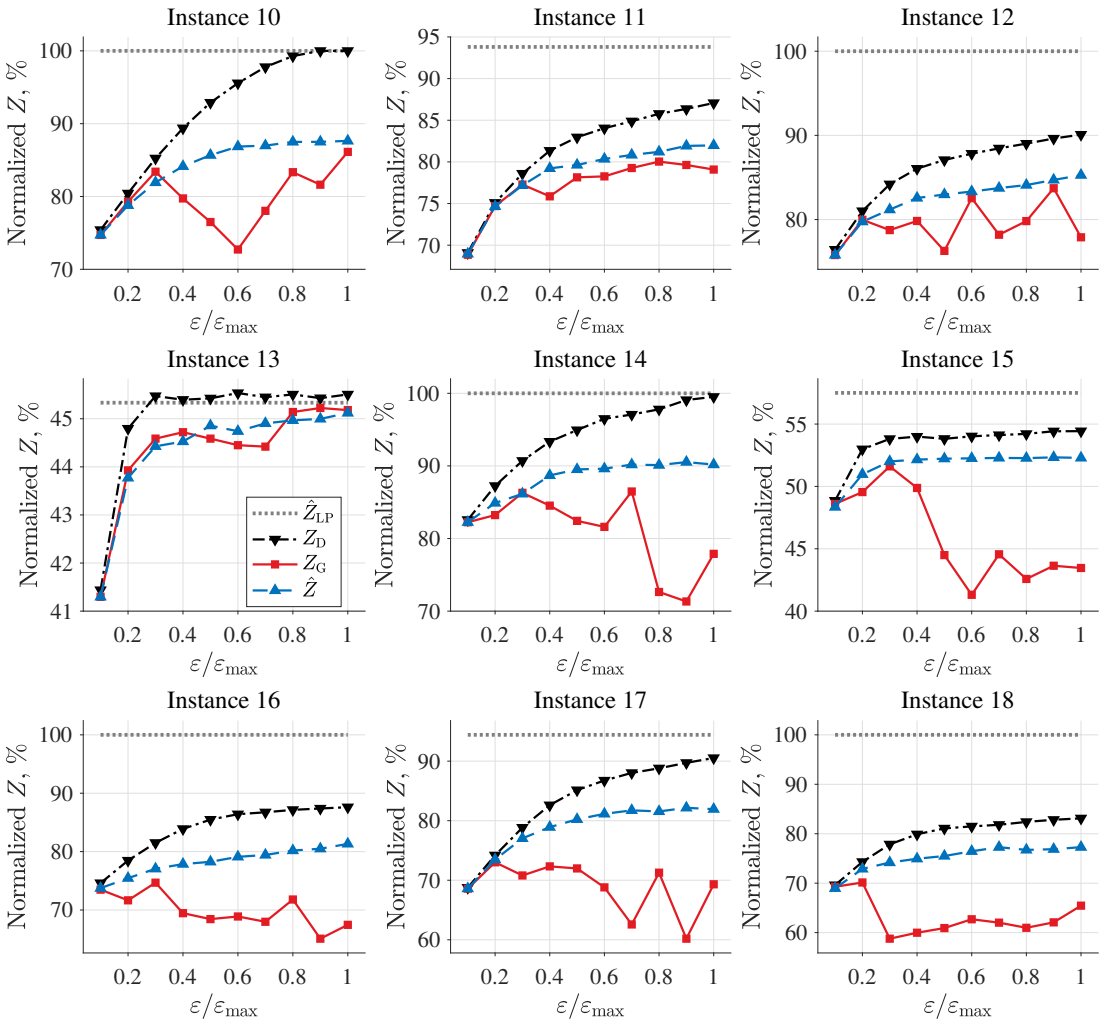


Figure 3.8: Computational results for instances 10–18. Note that all metrics are normalized and flipped in sign.

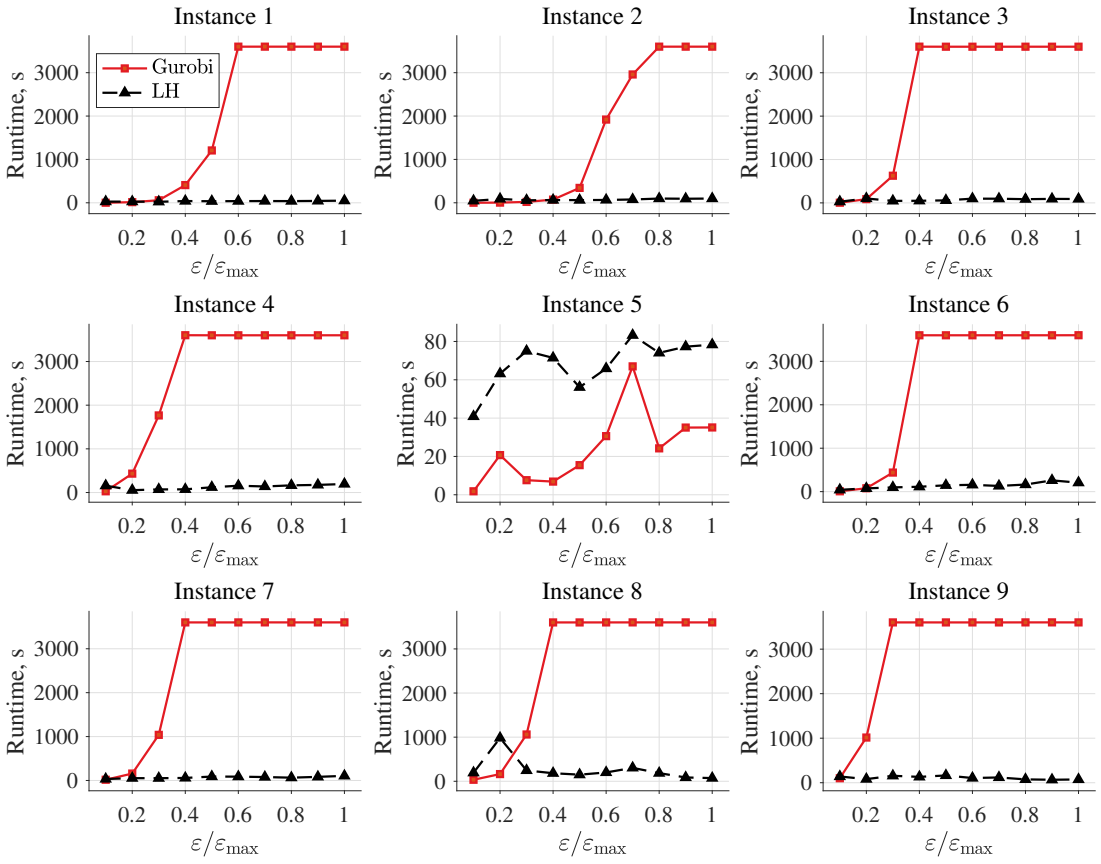


Figure 3.9: Runtime results for instances 1–9. The time limit of 3600 s is enforced.

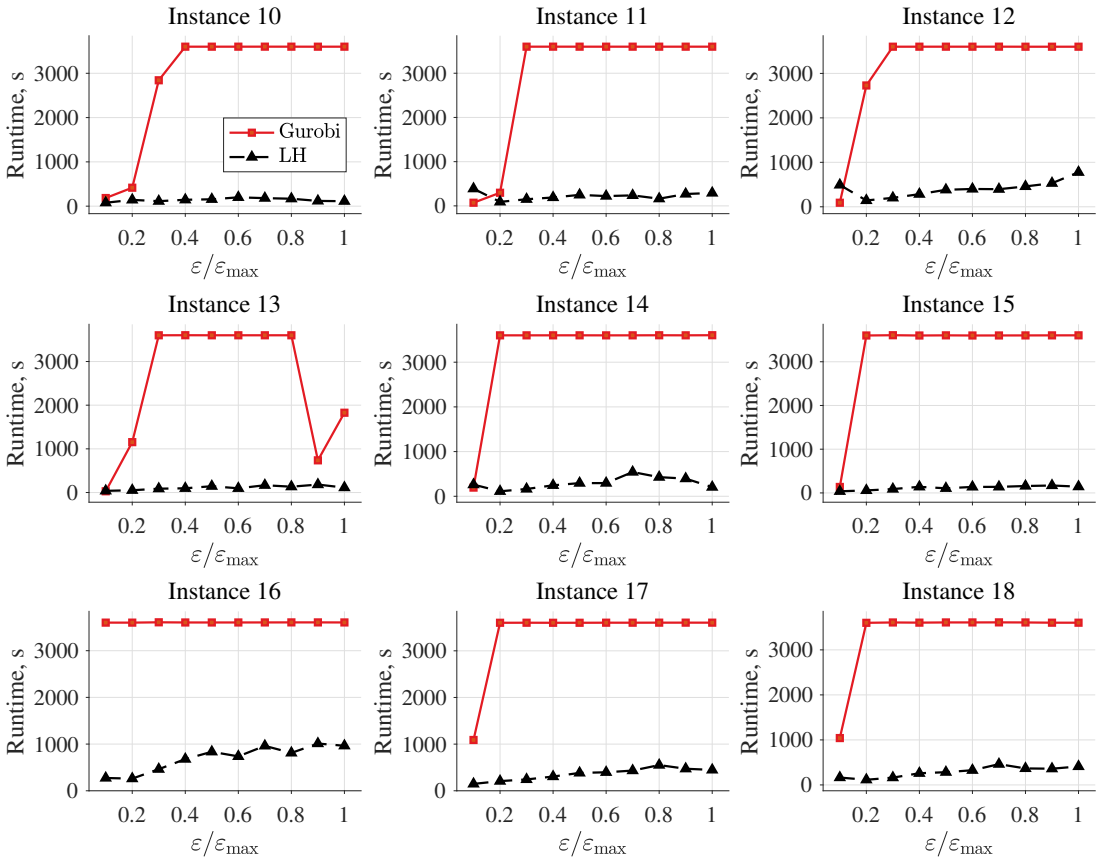


Figure 3.10: Runtime results for instances 10–18. The time limit of 3600 s is enforced.

applied it on all 180 instances of the computational experiments in Section 3.7.2. The results of the post-refinement procedure present that the maximum performance improvement provided by the post-refinement procedure is only 0.55 % for all 180 instances (see Figure 3.11). Thus, while the idea of the post-Lagrangian heuristic refinement procedure seems attractive, the results indicate the limited efficacy in improvements.

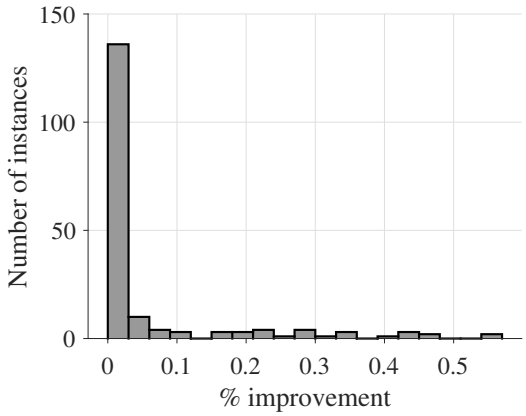


Figure 3.11: Improvements by the post-Lagrangian heuristic refinement.

3.7.4 Illustrative Example: Federated Disaster Monitoring

In this illustrative example, we focus on demonstrating three aspects: (i) the RCRP-IRC formulation in action, (ii) the Lagrangian heuristic method applied to RCRP-IRC, and (iii) non-RGT orbits.

Suppose a group of 7 satellites with different fuel states and heterogeneous orbits (parameters shown in the left half portion of Table 3.5) is tasked with a reconfiguration process to form a federation for satellite-based emergency mapping to monitor active disaster events and support post-disaster relief operations. The spot targets of interest are Getty, California (34.09°N, 118.47°S), Asheikri, Nigeria (11.96°N, 12.93°E), and Hunga Tonga–Hunga Ha'apai, Tonga (21.18°S, 175.19°W). We randomly generate the following parameters for this scenario. The size of the RCRP-IRC instance is $|\mathcal{I}| = 7$, $|\mathcal{J}| = 10,000$, $|\mathcal{T}| = 5,000$, and $|\mathcal{P}| = 3$. All 10,000 orbital slots are circular non-RGT orbits and have parameters $a \in [R_\oplus + 500, R_\oplus + 2500]$ where $R_\oplus$ is the mean radius of Earth, $e = 0$, $inc \in [20°, 85°]$,

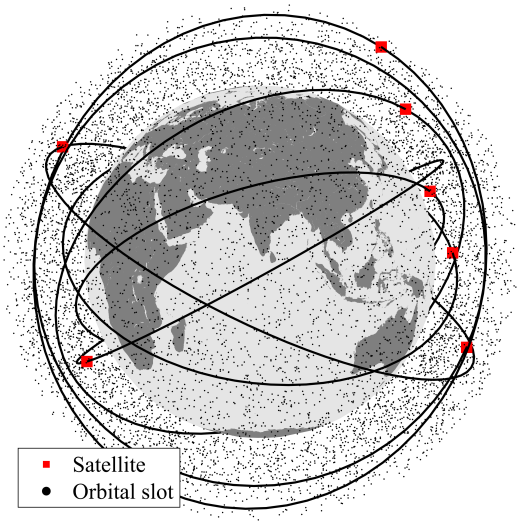
$\Omega \in [0°, 360°)$, and $u \in [0°, 360°)$. Figure 3.12a shows the positions of the satellites and the orbital slots at the epoch (J2000). We also add $|\mathcal{I}|$ additional slots to $\mathcal{J}$ to allow no maneuvering option for each satellite. Consequently, there are 70,049 assignment variables, 15,000 coverage state variables, and 25,021 constraints (excluding the decision variable domain definitions). To accommodate non-RGT orbits as candidate orbital slots in our framework, we need to define the time horizon for which the visibility profiles are considered valid (revisiting the generic case of the constellation-coverage model). For this, we set $T = 7$ days. It is important to note that due to the drifts of the orbital elements, the coverage persistency is not guaranteed beyond the specified simulation duration. The coverage rewards are randomly generated following the standard uniform distribution in the range of $[0, 1]$. For the cost matrix, we use the combined plane change and the Hohmann transfer maneuvers as outlined in Ref. [68]. The phasing angle is set to $\beta = \pi$ as the worst-case value; the corresponding Δv for the phasing serves as an upper value for the actual Δv required.

The initial configuration has the score of $Z = -1661.94$, which is 22.12 % of the total coverage reward possible. Solving RCRP-IRC using the Lagrangian heuristic method with the neighborhood size of $|\mathcal{N}'| \leq 50|\mathcal{I}|$, we obtain $\hat{Z} = -4017.70$, which is 53.48 % of the total coverage reward. The final configuration specification is shown in the right half of Table 3.5. The results indicate that all satellites raise their altitude to the maximum value, 9378.14 km, to maximize the coverage. All maneuvers are within the specified maximum Δv values. Figure 3.12 visualizes the initial and final configurations.

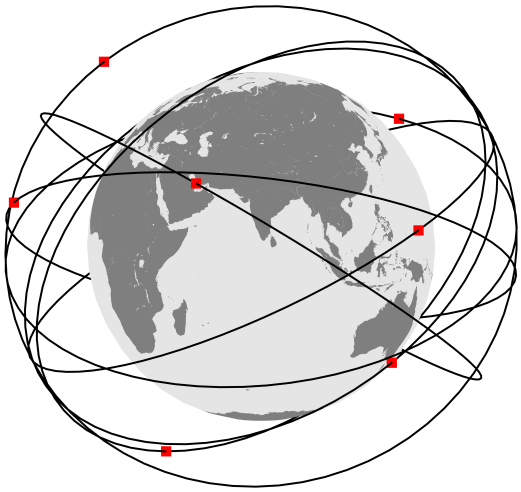
Solving the same instance of RCRP-IRC with the Gurobi optimizer (same setting as in Section 3.7.1), we obtain $Z_{\mathrm{G}} = -4077.03$, which is 54.27 % of the total coverage reward. The Lagrangian heuristic method terminates after 141.04 s while the Gurobi optimizer triggers the time limit of 3600 s. Out of 7 satellites, both the Lagrangian heuristic and the Gurobi optimizer return identical destination orbital slots for 5 satellites.

Table 3.5: Problem setting (left-half) and obtained Lagrangian heuristic solution (right-half) for the illustrative example.

Satellite	ε_i, km/s	Initial configuration				Final configuration (Solution)				Δv, km/s
		a, km	inc, deg.	Ω, deg.	u, deg.	a, km	inc, deg.	Ω, deg.	u, deg.	
1	2.79	8,236.65	73.68	79.09	338.41	9,375.47	63.14	81.27	130.46	2.19
2	2.60	7,574.06	28.89	352.30	294.35	9,363.32	26.52	17.60	19.53	2.46
3	2.06	7,939.43	57.38	292.21	121.00	9,371.60	58.18	301.59	274.92	1.97
4	4.80	8,990.08	77.94	61.90	63.15	9,369.90	45.11	53.49	328.51	4.65
5	4.12	6,889.94	33.60	293.84	134.22	9,374.35	22.78	249.93	29.86	3.76
6	3.40	7,182.06	32.05	98.67	2.05	9,373.86	27.04	75.66	67.33	2.61
7	3.67	8,555.01	27.04	155.41	90.87	9,369.79	32.87	196.13	140.21	3.26



(a) Initial configuration; shown in the figure are initial satellite orbits and orbital slots.



(b) Final configuration obtained by the Lagrangian heuristic.

Figure 3.12: Initial and final configurations.

3.8 Conclusions

This chapter proposes an integrated constellation design and transfer model for regional coverage constellation reconfiguration problem. Given a set of target points each associated with the time-varying coverage reward and the time-varying coverage threshold, the problem seeks to maximize the total reward obtained during the specified time horizon and to minimize the total cost of satellite transfers. The bi-objective formulation leads to the trade-off analysis (potentially the Pareto front analysis if all ε instances are solved to optimality) in the objective space spanned by the aggregated cost and the total coverage reward. Furthermore, as demonstrated in the illustrative example, the formulation is capable of accommodating different types of orbits, not necessarily restricting orbital slots to be RGT orbits. The use of non-RGT orbits require a user-specification of the time horizon T for which the formulation is valid for.

The ILP formulation of RCRP-ARC enables users to utilize the commercial software packages for convenient handling and tolerance-optimal solutions. However, for large-scale real-world instances, the problem suffers from the combinatorial explosion. To overcome this challenge and to produce high-quality feasible primal solutions, we developed the Lagrangian relaxation-based heuristic method that combines the subgradient method with the 1-exchange neighborhood local search, exploiting the special substructure of the problem. The computational experiments in Section 3.7 demonstrate the effectiveness of the proposed method particularly for large-scale instances, producing near-optimal solutions at significantly reduced computational runtime than the reference solver.

There are several interesting directions for future work in terms of modeling extensions and improvements in the solution approach. First regarding the modeling extensions, the assignment φ_{ij} of satellite i to orbit slot j can accommodate an additional *path* index to enable multi-path consideration. That is, a satellite can choose one from a set of paths to traverse to an orbital slot. Several different orbital transfer strategies can be considered

concurrently and hence lead to more flexible operations. A multi-stage reconfiguration can also be indeed studied with an appropriate set of assumptions, especially with regard to time constraints. Some of these interesting future research directions do not necessarily compromise the structure of the proposed problem but can significantly increase the size of problems. Inspired by this, the column generation (CG) approach can be applied based on the Dantzig-Wolfe decomposition (DWD) principle. The DWD-CG framework possesses several promising directions for future work, for example, lending itself to potentially efficient solution methods such as the restricted master heuristic (a CG-based heuristic), a combined LR-CG framework, and the branch-and-price algorithm. In addition, because the integrality gap is large for many instances, one can strengthen the Lagrangian relaxation bound by adding a set of valid inequalities.

We believe that the developed method provides an important step toward the realization of the concept of reconfiguration as a means for system adaptability and responsiveness, adding a new dimension to the operation of the next-generation satellite constellation systems.

This chapter is based on the following publication:

H. Lee and K. Ho, "Regional Constellation Reconfiguration Problem: Integer Linear Programming Formulation and Lagrangian Heuristic Method," *Journal of Guidance, Control, and Dynamics* (Submitted).

CHAPTER 4
MAXIMIZING OBSERVATION THROUGHPUT VIA MULTI-STAGE SATELLITE CONSTELLATION RECONFIGURATION

4.1 Introduction

Satellite constellation reconfiguration provides a space-borne system with flexibility and responsiveness in response to dynamic changes in mission requirements and environments. The concept of constellation reconfiguration has been explored in different application domains including the Earth observations (EO) [44, 53, 43], telecommunications [58], and navigation and positioning systems [47] for various reasons, spanning from the staged deployment to disaster monitoring. In this chapter, in the domain of Earth observations, we investigate the problem of reconfiguring a fleet of (potentially) heterogeneous satellites through multiple stages to maximize the observation rewards by achieving the coverage on targets of interest requested by clients.

One primary goal of Earth observations satellite systems is to maximize the system observational throughput. Prior studies have investigated the EO satellite scheduling problems (EOSSP) whose goal is to maximize the observation profit during a specified mission planning horizon while satisfying the complex operational constraints (e.g., solar panel charging, downlinking raw images) [83, 84]. In the classical EOSSP context, one of the underyling assumptions is that satellites point to their nadir directions without any attitude or maneuver controllability. Due to this assumption, the visible time windows (VTWs), which define the periods of satellite-to-target visibility, are considered fixed parameters to the scheduling problems. To improve the observational throughput, recent studies have explored the concept of "agile satellites" with attitude control capability [52, 85]. The agile satellites can control the orientation of their spacecraft body and directly manipulate the

VTWs within an EO scheduler. Therefore, the longer duration of the VTWs can be obtained, which in turn has the potential to enhance the overall observational throughput and scheduling efficiency.

We draw on the concept of satellite orbital transfer maneuverability as one of the most prominent notions of system flexibility along with satellite agility. Existing literature on satellite constellation reconfiguration has extensively focused on single-stage reconfiguration problem [69, 80] or without the consideration of optimizing a long mission planning horizon. However, constellation systems often face a series of reconfiguration opportunities arising due to satellite failures or change in mission objectives. Moreover, with the recent developments in on-orbit servicing, there is a greater potential to equip satellites with enhanced mobility for active orbital maneuvers [86]. Albeit these opportunities, several challenges also emerge. In practicing multi-stage reconfiguration, one of the main challenges that we confront is the excessive fuel consumption, particularly for high-thrust maneuvers in the low Earth orbit (LEO) regime. Therefore, it is critical to optimally lay out a set of orbital transfer paths of satellites through stages to maximize the profit of reconfiguration over a long-term mission planning horizon considering the fuel constraints.

In response to this background, we propose a novel integer linear (ILP) programming formulation for the multi-stage constellation reconfiguration problem (MCRP). MCRP is an extension to our prior work on single-stage reconfiguration problem [87]. Therefore, the formulation inherently features the heterogeneity in satellite hardware specifications and orbital characteristics and asymmetry in satellite distribution. The consideration of the heterogeneity is especially useful in modeling a cooperative EO missions such as disaster monitoring [50, 51]. The asymmetric satellite distribution can lead to efficient constellation pattern sets for EO applications as demonstrated in Ref. [59]. The problem can be solved using a state-of-the-art branch-and-bound algorithm for provably-optimal solutions.

The contribution of this chapter is two-fold:

1. **Multi-stage constellation reconfiguration problem.** We extend our prior work

on single-stage constellation reconfiguration problems using the basis of a time-expanded network. This modeling allows us to better understand the hidden design space that is otherwise overlooked with zero or single reconfiguration stage. The proposed model aims to concurrently optimize the design and transfer aspects of multiple reconfigurations over the entire mission planning horizon.

2. **Heuristic solution methods and empirical analysis.** We propose two sequential decision-making heuristic solution methods based on the principles of myopic policy and rolling horizon procedure to address the issue of computational intractability in solving large-scale problems. We empirically show that the myopic policy heuristic can be beneficial for instances with uniform observation rewards, and the rolling horizon procedure can be efficient for instances with dynamic environments.

The remainder of this chapter is organized as follows. In Section 4.2, we provide the formal description of the problem of multi-stage constellation reconfiguration and propose a novel ILP formulation. Section 4.3 discusses two heuristic solution methods. Section 4.4 then conducts computational experiments to compare the performances of the proposed methods on two sets of randomly-generated test instances. Lastly, in Section 4.5, we provide several interesting future work directions to enhance the applicability of the proposed work and conclude this chapter.

4.2 Multi-Stage Constellation Reconfiguration Problem

In this section, we describe and propose a mathematical optimization formulation of MCRP.

4.2.1 Problem Description

Given a finite discrete-time mission planning horizon $\mathcal{T} = \{1, 2, \ldots, T\}$, the objective of MCRP is to find a set of orbital transfer maneuver sequences of K heterogeneous satellites that maximizes the total obtained observation reward imposed on a set of target points $\mathcal{P}$.

Each satellite $k \in \mathcal{K}$ is associated with different hardware specifications (e.g., sensor field-of-view and propellant capacities) and orbital characteristics. In addition, each target point $p \in \mathcal{P}$ is associated with the non-negative time-dependent observation reward $\boldsymbol{\pi}_p = (\pi_{tp} \geq 0 : t \in \mathcal{T})$[1] and the minimum elevation angle threshold $\varepsilon_{\min}$. The observation reward π_{tp} is earned if at least r_{tp} number of satellites simultaneously cover target p at time step t. The concept of time-dependent observation reward models the "value" of the sensory data taken at a different time of day. One motivating example is the remote sensing application with visible spectral sensors; images taken under the Sun illumination may possess greater value than otherwise.

Time-Expanded Network

During the specified mission planning horizon, there are N *stages* at which a constellation can undergo reconfiguration processes. Denoting $\mathcal{S} = \{0, 1, \ldots, N\}$ by the set of stages (we let $s = 0$ indicate the initial state), we associate stage $s \in \mathcal{S}$ with the time-stamp $t_s \in \mathcal{T}$ and the stage planning horizon $\mathcal{T}_s = \{t : t_s \leq t < t_{s+1}, t \in \mathcal{T}\}$. We say that satellites simultaneously arrive at their new destination orbital slots at t_s to form a new constellation configuration. Without loss of generality, we assume that stage planning horizons evenly distribute the mission planning horizon.

The flows of satellites through stages are defined by a set of directed graphs $\{\mathcal{G}^1, \ldots, \mathcal{G}^K\}$ where we associate each satellite k with its own time-expanded graph (TEG) $\mathcal{G}^k = (\mathcal{J}^k, \mathcal{A}^k)$ as shown in the top part of Figure 4.1. Here, $\mathcal{J}^k = \{\mathcal{J}_0^k, \mathcal{J}_1^k, \ldots, \mathcal{J}_N^k\}$ is the set of the source node $\mathcal{J}_0^k$ and the time-expanded nodes $\{\mathcal{J}_1^k, \ldots, \mathcal{J}_N^k\}$ and $\mathcal{A}^k = \{\mathcal{A}_1^k, \ldots, \mathcal{A}_N^k\}$ is the set of arcs that connect the nodes of two adjacent stages. Here, each node set $\mathcal{J}_s^k$, $s \geq 1$ is a copy of $\mathcal{J}_1^k$ and has the cardinality J. The concept of TEGs allows us to model the time evolution of satellite states through stages over a set of identical nodes by associating each node with

[1]Note that our problem differs from the conventional problem settings considered in the EOSSP literature. In classical EOSSP, the images requested by the clients only need to be acquired once. The problem we are considering is the imaging of the same target points for a longer duration.

time-stamps. At each stage s, satellite k has options to either stay in its orbit or perform an active orbital maneuver to transfer from a prior stage's orbit $i \in \mathcal{J}_{s-1}^k$ to a new orbit $j \in \mathcal{J}_s^k$ with the non-negative cost of transfer $c_{ij} \geq 0$, which is deducted from the available resource $c_{s,\max}^k$. The source node of satellite k is included as an element of $\mathcal{J}_s^k, \forall s \in \mathcal{S}$ to enable the option to stay in orbit; consequently, $\exists c_{ij} = 0$ for $(i, j) \in \mathcal{A}_s^k, \forall s \in \mathcal{S} \setminus \{0\}, \forall k \in \mathcal{K}$.

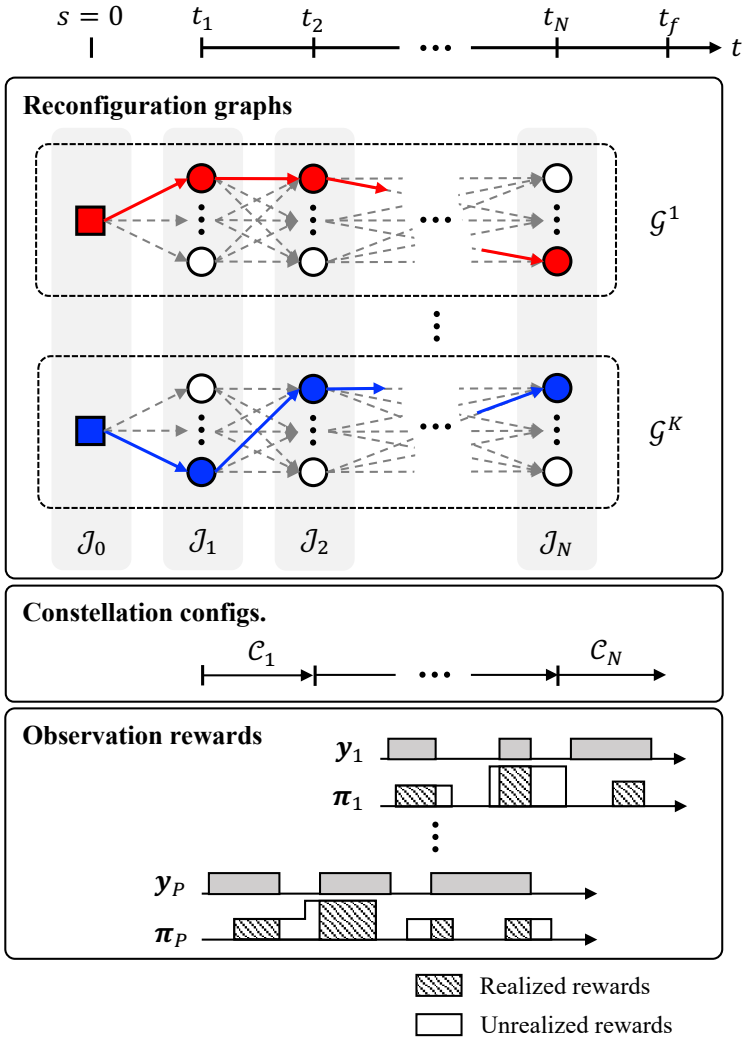


Figure 4.1: Mission planning horizon, reconfiguration graphs, constellation configurations, and observation rewards.

Observation Reward Mechanism

Each node is an orbital slot with the fixed coordinate in the Earth-centered inertial frame and is associated with the visibility profile per target point. Letting $x_j = 1$ to indicate the occupancy of orbital slot $j \in \mathcal{J}_s^k$ by satellite k at stage s ($x_j = 0$, otherwise), a set of newly occupied orbital slots forms a new constellation configuration $C_s := \{j : x_j = 1, j \in \mathcal{J}_s^k, k \in \mathcal{K}\}$ that is valid for the time interval $[t_s, t_{s+1})$ (see the middle part of Figure 4.1). We denote $\boldsymbol{y}_p = (y_{tp} \in \{0, 1\} : t \in \mathcal{T})$ by the VTW of target p where $y_{tp} = 1$ if target p is covered simultaneously by at least r_{tp} satellites ($y_{tp} = 0$, otherwise). The VTWs are the function of constellation configuration C_s during the stage planning horizon $\mathcal{T}_s$. Observation rewards are realized when VTWs are aligned with the periods of the non-negative observation rewards (see the bottom part of Figure 4.1). To obtain the maximal sum $\sum_{p\in\mathcal{P}} \sum_{t\in\mathcal{T}_s} \pi_{tp} y_{tp}$, the alignments of VTWs and the periods of high observation rewards need to be maximized.

Remarks

Each stage involves the optimization of (i) the *design* of a maximum-reward destination configuration and (ii) the minimum-cost *transfer* of satellites from one configuration to another. MCRP is an extension to the single-stage design-transfer problem explored in Ref. [87] by expanding it in the time dimension. In MCRP, all stages are coupled through the resource budget constraints. Therefore, no stage can be individually solved to obtain the true optimal solution to MCRP. If configuration designs in the early stages are aggressive in terms of resource consumption, then there will be no (or low) flexibility in designing good configurations in later stages. MCRP is about determining the optimal balance between the cost and the performance over the entire mission planning horizon. By increasing the degrees of freedom for reconfiguration throughout the mission planning horizon, the goal is to infuse more flexibility into the system and identify an optimal reconfiguration process that is otherwise overlooked with zero or single reconfiguration stage.

4.2.2 Mathematical Formulation

MCRP is a deterministic multi-period decision-making problem with the basis of network flows. However, the problem is not fully defined in a graph-theoretic setting because the reward set on a node is not a scalar value but involves conditional evaluation due to the linking between the constellation configuration and its coverage state [Constraints (4.2d)]. Therefore, the use of efficient algorithms for network flow problems such as the longest path problem cannot be applied.

To this end, we formulate MCRP as an ILP optimization problem. First, we define sets, parameters, and decision variables. Then, we introduce the mathematical formulation of MCRP.

Sets and indices

$\mathcal{S}$ Set of stage indices (index s; cardinality $N + 1$)

$\mathcal{K}$ Set of satellite indices (index k; cardinality K)

$\mathcal{J}_s^k$ Set of orbital slot indices of stage s for satellite k (indices i, j; cardinality J)

$\mathcal{P}$ Set of target point indices (index p; cardinality P)

$\mathcal{T}_s$ Planning horizon for stage s (index t)

$\mathcal{T}$ Mission planning horizon (index t; cardinality T)

Parameters

c_{ij} Cost of transferring satellite k from orbital slot $i \in \mathcal{J}_{s-1}^k$ to orbital slot $j \in \mathcal{J}_s^k$ ($c_{ij} \in \mathbb{R}_{\geq 0}$)

$c_{\max}^k$ Resource availability for satellite k

π_{tp} Coverage reward for target point p at time step t ($\pi_{tp} \in \mathbb{R}_{\geq 0}$)

r_{tp} Minimum coverage threshold to receive the reward of target point p at time step t ($r_{tp} \in \mathbb{Z}_{\geq 1}$)

V_{tjp} $\begin{cases} 1, & \text{if orbital slot } j \text{ is visible from target point } p \text{ at time step } t \\ 0, & \text{otherwise} \end{cases}$

Decision variables

φ_{ij} $\begin{cases} 1, & \text{if satellite } k \text{ transfers from orbital slot } i \in \mathcal{J}_{s-1}^k \text{ to orbital slot } j \in \mathcal{J}_s^k \\ 0, & \text{otherwise} \end{cases}$

y_{tp} $\begin{cases} 1, & \text{if target point } p \text{ is covered at time step } t \\ 0, & \text{otherwise} \end{cases}$

We denote $\mathbb{R}_{\geq 0}$ by the set of non-negative real numbers and $\mathbb{Z}_{\geq 1}$ by the set of integer numbers greater than or equal to one. We can describe the relationship between a flow on (i, j) with the destination node j as:

$$x_j = \sum_{i \in \mathcal{J}_{s-1}^k} \varphi_{ij}, \quad \forall j \in \mathcal{J}_s^k, \forall s \in \mathcal{S} \setminus \{0\}, \forall k \in \mathcal{K} \tag{4.1}$$

The mathematical formulation of MCRP for maximum observation throughput is as follows:

$$\text{(MCRP)} \quad \max \quad \sum_{p\in\mathcal{P}}\sum_{t\in\mathcal{T}} \pi_{tp} y_{tp} \tag{4.2a}$$

$$\text{s.t.} \quad \sum_{j\in\mathcal{J}_1^k} \varphi_{ij} = 1, \quad \forall i \in \mathcal{J}_0^k, \forall k \in \mathcal{K} \tag{4.2b}$$

$$\sum_{j\in\mathcal{J}_{s+1}^k} \varphi_{ij} - \sum_{q\in\mathcal{J}_{s-1}^k} \varphi_{qi} = 0, \quad \forall i \in \mathcal{J}_s^k, \forall s \in \mathcal{S}\setminus\{0,N\}, \forall k \in \mathcal{K} \tag{4.2c}$$

$$\sum_{k\in\mathcal{K}}\sum_{j\in\mathcal{J}_s^k}\sum_{i\in\mathcal{J}_{s-1}^k} V_{tjp}\varphi_{ij} \geq r_{tp} y_{tp}, \quad \forall t \in \mathcal{T}_s, \forall s \in \mathcal{S}\setminus\{0\}, \forall p \in \mathcal{P} \tag{4.2d}$$

$$\sum_{s\in\mathcal{S}\setminus\{0\}}\sum_{j\in\mathcal{J}_s^k}\sum_{i\in\mathcal{J}_{s-1}^k} c_{ij}\varphi_{ij} \leq c_{\max}^k, \quad \forall k \in \mathcal{K}' \subseteq \mathcal{K} \tag{4.2e}$$

$$\varphi_{ij} = \{0,1\}, \quad \forall i \in \mathcal{J}_{s-1}^k, \forall j \in \mathcal{J}_s^k, \forall s \in \mathcal{S}\setminus\{0\}, \forall k \in \mathcal{K} \tag{4.2f}$$

$$y_{tp} = \{0,1\}, \quad \forall t \in \mathcal{T}, \forall p \in \mathcal{P} \tag{4.2g}$$

The objective function (4.2a) maximizes the total reward (i.e., the total observation throughput) obtained by covering a set of target points of interest. Constraints (4.2b) are the initial stage outflow constraints. Constraints (4.2c) balances the outflow (the first term) and inflow (the second term) of the nodes of intermediate stages. Constraints (4.2d) are the configuration-coverage linking constraints that ensure that target point p is covered at time step t only if there exists at least r_{tp} satellite(s) in view. Constraints (4.2d) couples the flow of satellites at every stage. Constraints (4.2e) are the resource availability constraints that restrict the maximum allowable Δv of satellite k to $c_{\max}^k$. The set $\mathcal{K}' \subseteq \mathcal{K}$ is used to denote the subset of satellites that impose such resource availability constraints. Constraints (4.2f)–(4.2g) define the domain of decision variables.

4.3 Heuristic Methods

In Section 4.2, we formulated MCRP as an integer linear program. Consequently, we can utilize generic mixed-integer linear programming (MILP) methods such as the branch-and-bound algorithm to solve the problem. However, MCRP is a combinatorial optimization problem that suffers from the curse of dimensionality as the total number of potentially feasible plans grows exponentially with the linear increase in J, N, and K (e.g., there are at most J^{NK} plans to consider). For example, an instance I of MCRP with 3 reconfiguration stages, 5 satellites, and 50 candidate orbital slots per satellite has up to 3.05×10^{25} potentially feasible plans. The enumeration of these plans for feasibility and optimality checks can be computationally prohibitive.

To address the computational intractability in solving MCRP, we construct two sequential decision-making heuristic methods based on the principles of myopic policy and the rolling horizon procedure [88]. Feasible solutions obtained by the heuristic methods are feasible solutions to MCRP.

4.3.1 Myopic Policy Heuristic

To circumvent the challenge of combinatorial explosion, we develop a divide-and-conquer sequential decision-making framework called the *Myopic Policy Heuristic* (MPH). The principal idea is to partition MCRP by stages into N smaller subproblems with manageable sizes and solve subproblems in a successive manner. With the knowledge of the satellite states from the precedent stage, which we denote with $\tilde{\mathcal{J}}_{s-1}$, the number of potentially feasible plans effectively reduces to J^K per subproblem. Considering the same instance I of MCRP, we can partition the problem into 3 subproblems; each subproblem has up to 3.13×10^{8} potentially feasible plans. Small subproblems can be efficiently solved using a commercial software package. Nevertheless, additional algorithmic efforts can be applied on the basis of MPH to further improve the performance such as the solution quality and

the time complexity. It is important to note that MPH is an algorithmic framework with a myopic policy. That is, no impacts on the future stages are considered in the current-stage decision-making. Figure 4.2 illustrate the scope of a subproblem.

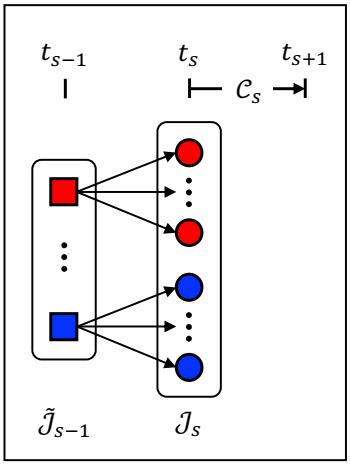


Figure 4.2: MPH subproblem for stage s.

Subproblem Formulation

A subproblem is parameterized with the stage index s and is denoted with $\mathsf{SP}(s)$. The mathematical formulation of $\mathsf{SP}(s)$ is as follows:

$$(\mathsf{SP}(s)) \quad z_s = \max \quad \sum_{p\in\mathcal{P}} \sum_{t\in\mathcal{T}_s} \pi_{tp} y_{tp} \tag{4.3a}$$

$$\text{s.t.} \quad \sum_{j\in\mathcal{J}_s^k} \varphi_{ij} = 1, \quad \forall i \in \tilde{\mathcal{J}}_{s-1}^k, \forall k \in \mathcal{K} \tag{4.3b}$$

$$\sum_{i\in\tilde{\mathcal{J}}_{s-1}^k} \varphi_{ij} \leq 1, \quad \forall j \in \mathcal{J}_s^k, \forall k \in \mathcal{K} \tag{4.3c}$$

$$\sum_{k\in\mathcal{K}} \sum_{j\in\mathcal{J}_s^k} \sum_{i\in\tilde{\mathcal{J}}_{s-1}^k} V_{tjp}\varphi_{ij} \geq r_{tp} y_{tp}, \quad \forall t \in \mathcal{T}_s, \forall p \in \mathcal{P} \tag{4.3d}$$

$$\sum_{j\in\mathcal{J}_s^k} \sum_{i\in\tilde{\mathcal{J}}_{s-1}^k} c_{ij}\varphi_{ij} \leq c_{s,\max}^k, \quad \forall k \in \mathcal{K}' \subseteq \mathcal{K} \tag{4.3e}$$

$$\varphi_{ij} = \{0,1\}, \quad \forall i \in \tilde{\mathcal{J}}_{s-1}^k, \forall j \in \mathcal{J}_s^k, \forall k \in \mathcal{K} \tag{4.3f}$$

$$y_{tp} = \{0,1\}, \quad \forall t \in \mathcal{T}_s, \forall p \in \mathcal{P} \tag{4.3g}$$

The objective function (4.3a) maximizes the total observation reward for stage s. Constraints (4.3b) and (4.3c) are the usual assignment problem constraints. Constraints (4.3d) are the configuration-coverage linking constraints for stage s. Constraints (4.3e) are the individual resource availability constraints; unlike Constraints (4.2e), the resource availability constraints reflect the resource consumptions from the prior stages $\{1, \ldots, s-1\}$ and parameterize them; it is computed as follows:

$$c_{s,\max}^k = c_{\max}^k - \sum_{q=1}^{s-1} \sum_{j\in\mathcal{J}_q^k} \sum_{i\in\tilde{\mathcal{J}}_{q-1}^k} c_{ij}\varphi_{ij} \tag{4.4}$$

Constraints (4.3f) and (4.3g) define the domain of decision variables.

Algorithm 5 outlines the overall solution procedure of MPH. $\mathsf{SP}(s)$ outputs the optimum z_s and the optimal assignment solution $\boldsymbol{\varphi}_s^* = (\varphi_{ij}^* = \{0,1\}, i \in \tilde{\mathcal{J}}_{s-1}^k, j \in \mathcal{J}_s^k, k \in \mathcal{K})$ and the

optimal coverage state solution $\boldsymbol{y}_s^* = (y_{tp}^* \in \{0,1\} : t \in \mathcal{T}_s, p \in \mathcal{P})$. (Notice that the term optimality is with respect to SP(s)). The algorithm stores the results from each stage and returns the heuristic solution objective value z_{mph}, which is the sum of all z_s, and a feasible solution $(\boldsymbol{\varphi}^*, \boldsymbol{y}^*)$, which is the collection of all $(\boldsymbol{\varphi}_s^*, \boldsymbol{y}_s^*)$, to MCRP.

Algorithm 5: Myopic policy heuristic

Input: $\boldsymbol{c}$, $\boldsymbol{\pi}$, $\boldsymbol{V}$, $\boldsymbol{r}$
Output: z_{mph}, $(\boldsymbol{\varphi}^*, \boldsymbol{y}^*)$

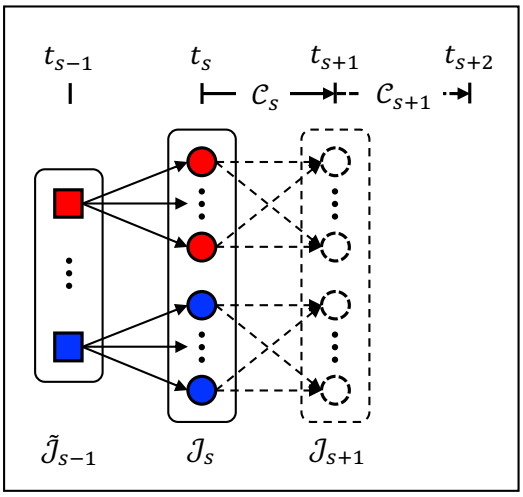
Initialize $s \leftarrow 1$
Compute SP(s) and store: z_1 and $(\boldsymbol{\varphi}_1^*, \boldsymbol{y}_1^*)$
$s \leftarrow s + 1$
while $s \leq N$ **do**
    Update $\tilde{\mathcal{J}}_{s-1}^k \leftarrow \{j : \varphi_{ij} = 1, i \in \mathcal{J}_{s-2}^k, j \in \mathcal{J}_{s-1}^k\}$
    Compute SP(s) and store: z_s and $(\boldsymbol{\varphi}_s^*, \boldsymbol{y}_s^*)$
    $s \leftarrow s + 1$
$z_{\text{mph}} \leftarrow \sum_{s \in \mathcal{S} \setminus \{0\}} z_s$
$(\boldsymbol{\varphi}^*, \boldsymbol{y}^*) \leftarrow ((\boldsymbol{\varphi}_s^*, \boldsymbol{y}_s^*) : s \in \mathcal{S} \setminus \{0\})$

The formulation of SP(s) is identical to the regional constellation reconfiguration problem with individual resource constraints (RCRP-IRC) as outlined in Ref. [87]. The problem embeds a budgeted assignment problem and a maximal covering location problem. As discussed previously, any dedicated algorithm can be applied to solve SP(s). In our case, we can exploit the established Lagrangian-relaxation based heuristic method for RCRP-IRC.

4.3.2 Rolling Horizon Procedure

The *Rolling Horizon Procedure* (RHP) uses the impact of the current-stage decisions on future stages to make informed decisions at the current stage [89]. In this chapter, we use the deterministic 1-stage lookahead policy. This allows us to partition MCRP into $N - 1$ smaller subproblems; the last iteration at $s = N - 1$ deterministically optimizes the entire remaining mission planning horizon. Due to the lookahead policy, each subproblem is larger than the subproblem SP(s) of MPH. Considering the same instance I of MCRP, we can partition the problem into 2 subproblems; each subproblem has up to 9.76×10^{16}

potentially feasible plans. Figure 4.3 illustrates the scope of a subproblem with the 1-stage lookahead policy.



1-SP(s)

Figure 4.3: RHP subproblem for stage s with the 1-stage lookahead policy.

Subproblem Formulation

We denote 1-SP(s) by a subproblem parameterized with the stage index s and the 1-stage lookahead policy. The mathematical formulation of 1-SP(s) is as follows:

$$(\text{1-SP}(s)) \quad z_s + z_{s+1} = \max \quad \sum_{p\in\mathcal{P}} \sum_{t\in\{\mathcal{T}_s,\mathcal{T}_{s+1}\}} \pi_{tp} y_{tp} \tag{4.5a}$$

$$\text{s.t.} \quad \sum_{j\in\mathcal{J}_s^k} \varphi_{ij} = 1, \quad \forall i \in \tilde{\mathcal{J}}_{s-1}^k, \forall k \in \mathcal{K} \tag{4.5b}$$

$$\sum_{j\in\mathcal{J}_{s+1}^k} \varphi_{ij} - \sum_{q\in\tilde{\mathcal{J}}_{s-1}^k} \varphi_{qi} = 0, \quad \forall i \in \mathcal{J}_s^k, \forall k \in \mathcal{K} \tag{4.5c}$$

$$\sum_{k\in\mathcal{K}} \sum_{j\in\mathcal{J}_s^k} \sum_{i\in\tilde{\mathcal{J}}_{s-1}^k} V_{tjp}\varphi_{ij} \geq r_{tp} y_{tp}, \quad \forall t \in \mathcal{T}_s, \forall p \in \mathcal{P} \tag{4.5d}$$

$$\sum_{k\in\mathcal{K}} \sum_{j\in\mathcal{J}_{s+1}^k} \sum_{i\in\mathcal{J}_s^k} V_{tjp}\varphi_{ij} \geq r_{tp} y_{tp}, \quad \forall t \in \mathcal{T}_{s+1}, \forall p \in \mathcal{P} \tag{4.5e}$$

$$\sum_{j\in\mathcal{J}_s^k} \sum_{i\in\tilde{\mathcal{J}}_{s-1}^k} c_{ij}\varphi_{ij} + \sum_{j\in\mathcal{J}_{s+1}^k} \sum_{i\in\mathcal{J}_s^k} c_{ij}\varphi_{ij} \leq c_{s,\max}^k, \quad \forall k \in \mathcal{K}' \subseteq \mathcal{K} \tag{4.5f}$$

$$\varphi_{qi}, \varphi_{ij} = \{0, 1\}, \quad \forall q \in \tilde{\mathcal{J}}_{s-1}^k, \forall i \in \mathcal{J}_s^k, \forall j \in \mathcal{J}_{s+1}^k, \forall k \in \mathcal{K} \tag{4.5g}$$

$$y_{tp} = \{0, 1\}, \quad \forall t \in \{\mathcal{T}_s, \mathcal{T}_{s+1}\}, \forall p \in \mathcal{P} \tag{4.5h}$$

The objective function (4.5a) maximizes the sum of observation rewards for stages s and $s+1$. Constraints (4.5b) and (4.5c) are the usual network flow conservation constraints. Constraints (4.5d) and (4.5e) are the configuration-coverage linking constraints for stages s and $s+1$, respectively. Constraints (4.5f) are the individual resource availability constraints; $c_{s,\max}^k$ is defined in Eq. (4.4). Constraints (4.5g) and (4.5h) define the domain of decision variables.

Algorithm 6 overviews RHP. In essence, the structure of this algorithm is very similar to that of Algorithm 5 but it possesses one distinct characteristic. The arguments of 1-

SP(s) are $(\boldsymbol{\varphi}^*_s, \boldsymbol{y}^*_s)$ and $(\boldsymbol{\varphi}^*_{s+1}, \boldsymbol{y}^*_{s+1})$. However, we do not skip stage +1 and proceed directly to stage $s + 2$. Instead, we only make decisions for the stage s using the deterministic forecast of the next immediate stage and proceed to stage $s + 1$. As discussed previously, no decisions are made for stage $s + 1$ except if the current stage is at $N - 1$.

Algorithm 6: Rolling horizon procedure

Input: $\boldsymbol{c}, \boldsymbol{\pi}, \boldsymbol{V}, \boldsymbol{r}$
Output: $z_{\text{rhp}}, (\boldsymbol{\varphi}^*, \boldsymbol{y}^*)$

Initialize $s \leftarrow 1$
Compute 1-SP(s) and store: z_1 and $(\boldsymbol{\varphi}^*_1, \boldsymbol{y}^*_1)$
$s \leftarrow s + 1$
while $s \leq N - 1$ **do**
    Update $\tilde{\mathcal{J}}^k_{s-1} \leftarrow \{j : \varphi_{ij} = 1, i \in \mathcal{J}^k_{s-2}, j \in \mathcal{J}^k_{s-1}\}$
    if $s < N - 1$ **then**
        Compute 1-SP(s) and store: z_s and $(\boldsymbol{\varphi}^*_s, \boldsymbol{y}^*_s)$
    else
        Compute 1-SP(s) and store: $\{z_s, z_{s+1}\}$ and $((\boldsymbol{\varphi}^*_s, \boldsymbol{y}^*_s), (\boldsymbol{\varphi}^*_{s+1}, \boldsymbol{y}^*_{s+1}))$
    $s \leftarrow s + 1$
$z_{\text{rhp}} \leftarrow \sum_{s \in \mathcal{S} \setminus \{0\}} z_s$
$(\boldsymbol{\varphi}^*, \boldsymbol{y}^*) \leftarrow ((\boldsymbol{\varphi}^*_s, \boldsymbol{y}^*_s) : s \in \mathcal{S} \setminus \{0\})$

4.3.3 Upper Bound of MCRP for Heuristic Solution Gap Analysis

The principal motivation for us to consider MPH and RHP is to address the issue of the curse of dimensionality in solving MCRP. While these heuristic methods can compute feasible solutions to MCRP relatively faster than concurrently solving for the entire mission planning horizon, there is no guarantee of the optimality of the obtained solutions.

To analyze the quality of a generic heuristic solution z_{h} without the knowledge of the optimal solution z, we identify an upper bound $\hat{z}$ to MCRP that can be used to compute the duality gap (DG), that is, the difference between the upper bound of MCRP and the heuristic solution. Ensuring that the optimal solution is always bounded between the upper bound and the heuristic solution, we can use such information to infer the quality of the heuristic solution with respect to the unknown optimal solution. One typical upper bound

for this purpose is the solution to the LP relaxation problem of MCRP. However, obtaining the LP relaxation bound can also be computationally challenging in large-scale instances. In what follows, we describe one upper bound metric that can be computed with a given set of parameters.

To find an upper bound $\hat{z}$ of MCRP, we begin by relaxing Constraints (4.2e). This decouples the coupling between stages. Also, this proves that any upper bound metric derived beyond this point will always satisfy $\hat{z} \geq z$, and hence proves the boundness of the optimum.

Solving the resource availability-relaxed MCRP can be still challenging because the problem now consists of N maximum coverage problems, which are shown to be NP-hard [87, 71]. In what follows, we present a computationally-efficient upper bound metric.

We begin by examining Constraints (4.2d) for stage s and aggregating $t \in \mathcal{T}_s$ and $p \in \mathcal{P}$:

$$\sum_{p\in\mathcal{P}} \sum_{t\in\mathcal{T}_s} \frac{\pi_{tp}}{r_{tp}} \sum_{k\in\mathcal{K}} \sum_{j\in\mathcal{J}_s^k} \sum_{i\in\mathcal{J}_{s-1}^k} V_{tjp}\varphi_{ij} \geq \sum_{p\in\mathcal{P}} \sum_{t\in\mathcal{T}_s} \pi_{tp} y_{tp}$$

With this, we wish to maximize the left-hand side. We can do so by casting it as a maximization problem:

$$\hat{z}_s = \max_{x_j\in\{0,1\}, j\in\mathcal{J}_s^k, k\in\mathcal{K}} \left\{ \sum_{p\in\mathcal{P}} \sum_{t\in\mathcal{T}_s} \frac{\pi_{tp}}{r_{tp}} \sum_{k\in\mathcal{K}} \sum_{j\in\mathcal{J}_s^k} V_{tjp} x_j : \sum_{j\in\mathcal{J}_s^k} x_j = 1, k \in \mathcal{K} \right\}$$

where we use Eq. (4.1) to change variables. This problem can be further decomposed into K subproblems, each with the satellite index $k \in \mathcal{K}$ as a parameter:

$$\hat{z}_s^k = \max_{x_j\in\{0,1\}, j\in\mathcal{J}_s^k} \left\{ \sum_{p\in\mathcal{P}} \sum_{t\in\mathcal{T}_s} \frac{\pi_{tp}}{r_{tp}} \sum_{j\in\mathcal{J}_s^k} V_{tjp} x_j : \sum_{j\in\mathcal{J}_s^k} x_j = 1 \right\}$$

By aggregating $\hat{z}_s$ for all stages, we obtain an upper bound of MCRP:

$$\hat{z} = \sum_{s \in \mathcal{S} \setminus \{0\}} \hat{z}_s$$

4.4 Computational Experiments

In the first part of this section, we conduct computational experiments to evaluate the impact of the problem size on the quality of the heuristic solutions obtained by MPH and RHP compared to the baseline MCRP. All problems are solved using a commercial MILP solver. The second part of this section conducts a detailed analysis to investigate the impact of having multiple stages on the system observational throughput.

4.4.1 Comparative Analysis

Experimental Setup

We have two sets of uniquely-generated test instances. Each set consists of 12 test instances, and each test instance draws one parameter value from each of the following sets: $N \in \{3, 4, 5\}$, $J \in \{50, 150\}$, and $K \in \{3, 5\}$. The smallest instance has at most 1.96×10^{15} feasible plans, and the largest instance has at most 2.53×10^{54} feasible plans. The goal is to investigate the impact of the problem size on the solution quality and the computational runtime. For each set of test instances, we apply a different parameter generation rule; no two instances with the same problem dimension have identical parameters.

For the first set, we set following parameter values. We assume a group of K homogeneous satellites in inclined circular orbits following the Walker-δ constellation pattern rule of $80^\circ : K/K/0$. This indicates that each satellite occupies its own orbital plane and the relative phasing between satellites in adjacent orbital planes is zero. The altitude of the constellation system is randomly selected in the range [700 km, 2000 km]. The set of $P = 10$ spot targets are randomly generated in the latitude interval $[-80^\circ, 80^\circ]$ and no

restriction is set on the longitudes; all spot targets are set with $\varepsilon_{\min} = 5^\circ$. We assume $\pi_{tp} = 1, \forall t \in \mathcal{T}, p \in \mathcal{P}$ and $r_{tp} = 1, \forall t \in \mathcal{T}, p \in \mathcal{P}$. This models that all targets have identical weights for imaging. We let the resource availability to $c^k_{\max} = 600\,\mathrm{m/s}, \forall k \in \mathcal{K}$. The considered mission planning horizon is for 5 days and it is discretized with the time step size of $t_{\text{step}} = 100\,\mathrm{s}$; consequently, we have $T = 4320$. Each $\mathcal{J}^k_s = \{1, \ldots, J\}$ is generated such that each orbital slot is associated with identical orbital elements but different true anomaly; the true anomalies of orbital slots within the orbital plane of satellite k is uniformly spaced.

For the second set, we focus on analyzing the impacts of the spatiotemporally-varying observation rewards on the quality of solutions. We keep every parameter generation rules the same as the first set, but we vary the following parameter values. We assume that there are different sets of targets $\{\mathcal{P}_1, \ldots, \mathcal{P}_N\}$ that are valid during the period of a particular stage, and each target $p \in \mathcal{P}_s$ is associated with the temporally-varying reward following the rule:

$$\pi_{tp} = \begin{cases} \sim U(0,1), & \text{if } t \in \mathcal{T}_s \\ 0, & \text{otherwise} \end{cases}$$

where $U(0, 1)$ is the uniform distribution in $[0, 1]$. We set no observation rewards on targets beyond or before the periods of the assigned stages. We intend to simulate dynamically-changing environments such that a constellation configuration optimized for one stage would be drastically unfit for another. This setting is in contrast to that of the first set because all targets have uniform rewards throughout the entire mission planning horizon and no targets are dynamically generated or removed.

An arc (i, j) is associated with the cost of transfer c_{ij}. In our problem domain, the cost is the Δv required to transfer a satellite from one orbital slot to another. Given the many-maneuver opportunistic nature of multi-stage reconfiguration, it is logical to assume only co-planar maneuvers in this chapter. More specifically, we restrict the co-planar maneuvers to phasings only. Out-of-plane impulsive maneuvers are especially costly in the LEO

regime; therefore, such maneuvers are not ideal for a series of orbital transfers. We approximate the cost of transferring satellite k from orbital slot $i \in \mathcal{J}_{s-1}^k$ to orbital slot $j \in \mathcal{J}_s^k$ at stage s by taking these two nodes as the boundary conditions of a circular co-planar phasing problem as outlined in Ref. [68].

We utilize a commercial software package, the Gurobi optimizer (version 9.1.1.), to solve MCRP and the subproblems of MPH and RHP. All computational experiments are coded and conducted on a platform with the Intel Core i-9700 3.00 GHz CPU processor (8 cores and 8 threads) and 32 GB of memory. In all cases, we allow the Gurobi optimizer to utilize all available cores. We use the default parameters of the Gurobi optimizer except that we impose the runtime limit of 3600 s. No early termination is enforced on the heuristic methods even if the runtime aggregated thus far exceeds 3600 s.

To gauge the quality of heuristic solutions relative to the MCRP solution obtained by the Gurobi optimizer, we define the *relative performance metric* (RP): $(z_\mathrm{h} - z)/z_\mathrm{h}$ unrestricted in sign where z_h denotes the generic heuristic solution objective function value. The positive sign of RP indicates the outperformance of a heuristic method relative to the Gurobi optimizer for MCRP. In cases where computing the optimal solutions of MCRP is computationally prohibitive, we can infer the quality of the heuristic solutions by computing the duality gap that bounds the optimal solution. We define DG as follows: $|\hat{z} - z_\mathrm{h}|/z_\mathrm{h}$. Note that the LP relaxation solution of MCRP can be used in place of $\hat{z}$, but quantifying it can be computationally challenging.

Numerical Results

The top part of Table 4.1 (instances 1–12) reports the results of the computational experiments on the first set of test instances. Out of 12 MCRP instances, the Gurobi optimizer triggers the time limit of 3600 s on 9 instances and optimally solves 3 instances. For instance 11, the Gurobi optimizer forcefully terminates due to the out-of-memory issue. For the same instance, however, both MPH and RHP manage to obtain feasible solutions. In

particular, MPH solves instance 11 with the duality gap of 2.98 % in 370.26 s whereas RHP solves the same instance with the duality gap of 11.27 % in 4770.89 s (the first stage subproblem triggers the time limit). Overall, solving MCRP, MPH, and RHP using the Gurobi optimizer led to finding 4, 4, and 6 best solutions, respectively (see boldface entries). In instances in which MPH underperforms relative to MCRP, the relative under-performance is at most 0.76 %. A similar analysis extends to RHP with the relative underperformance with at most 0.31 %. RHP found more high-quality solutions than others, however, RHP comes at the price of additional computational runtime. For instances 11 and 12, which represent "large-scale" problems, RHP achieves solutions with relatively larger DG than other instances, and we can observe that MPH performs better than RHP with much lower DGs.

The bottom part of Table 4.1 (instances 13–24) reports the results of the computational experiments on the second set of test instances. Out of 12 instances, MCRP formulation led to finding the best solutions on 8 instances. As described previously, the test instances of the second set have drastic changes in the targets of interest that vary stage by stage. Therefore, the ability to concurrently optimize the entire stage is highly desired. In addition to this empirical result, we wish to see how MPH and RHP would perform in such scenarios by comparing the metrics such as RPs and DGs. MPH and RHP found 0 and 6 best solutions, respectively. In general, the duality gaps of heuristic methods are poorer than those we computed in the first set. Examining the relative performance metrics, we can see that the worst underperformance of MPH and RHP relative to MCRP are 4.68 % and 1.37 %, respectively. These values are poorer than what we found in the first set of instances, however, we believe that these numbers indicate the high-quality solutions of the heuristic methods, especially considering the computational runtime required to produce such results. In cases we cannot quantify z, we need to resort to duality gaps to infer the solution quality. We observe that the metric $\hat{z}$ provides a good upper bound estimate of the optimum to MCRP.

Table 4.1: Results from the computational experiments. The boldface fonts indicate the maximum of z, z_{mph}, and z_{rhp}.

Instance				MCRP				MPH				RHP				No recon.
ID	J	N	K	z	Runtime,* s	DG†, %	$\hat{z}$	z_{mph}	Runtime, s	RP, %	DG, %	z_{rhp}	Runtime, s	RP, %	DG, %	
1	50	3	3	**12,018**	1,835.34	0.00	12,159	**12,018**	2.46	0.00	1.17	11,981	52.08	-0.31	1.49	10,213
2	50	3	5	13,582	-	20.00	16,473	15,871	64.53	14.42	3.79	**15,947**	3,608.29‡	14.83	3.30	12,117
3	50	4	3	**7,328**	200.54	0.00	7,440	7,273	0.36	-0.76	2.30	7,307	16.21	-0.29	1.82	5,745
4	50	5	3	**12,477**	-	0.27	12,591	12,444	1.62	-0.27	1.18	12,476	32.13	-0.01	0.92	8,852
5	50	4	5	13,473	-	18.04	16,108	15,239	107.59	11.59	5.70	**15,249**	3,754.84‡	11.65	5.63	12,224
6	150	3	3	**5,408**	1,103.81	0.00	5,447	5,371	3.78	-0.69	1.42	**5,408**	86.47	0.00	0.72	4,713
7	50	5	5	16,043	-	26.47	20,643	19,802	64.42	18.98	4.25	**19,803**	3,625.95‡	18.99	4.24	12,198
8	150	3	5	7,730	-	2.04	8,003	**7,832**	27.73	1.30	2.18	7,746	3,634.44‡	0.21	3.32	6,597
9	150	4	3	9,391	-	5.44	9,991	9,836	20.00	4.52	1.58	**9,840**	466.63	4.56	1.53	8,066
10	150	5	3	10,775	-	0.87	11,015	10,808	17.66	0.31	1.92	**10,823**	328.39	0.44	1.77	8,331
11§	150	4	5	N/A	-	N/A	24,050	**23,354**	370.26	N/A	2.98	21,615	4,770.89‡	N/A	11.27	13,878
12	150	5	5	16,118	-	167.96	20,526	**19,659**	1,573.23	18.01	4.41	18,187	3,667.17‡	11.38	12.86	12,989
13	50	3	3	**2,458.23**	693.75	0.00	2,539.16	2,428.22	1.56	-1.24	4.57	**2,458.23**	25.41	0.00	3.29	1,758.90
14	50	3	5	3,357.83	-	7.63	3,744.99	3,343.33	5.94	-0.43	12.01	**3,456.53**	3,732.53‡	2.86	8.35	2,378.22
15	50	4	3	**2,172.80**	-	1.09	2,266.15	2,162.64	0.55	-0.47	4.79	2,157.82	19.36	-0.69	5.02	1,620.92
16	50	5	3	**582.51**	7.68	0.00	607.38	556.49	0.12	-4.68	9.14	**582.51**	2.56	0.00	4.27	490.52
17	50	4	5	**1,918.26**	-	0.61	2,024.75	1,852.82	1.43	-3.53	9.28	1,915.98	18.34	-0.12	5.68	1,202.84
18	150	3	3	**2,059.32**	3,341.73	0.00	2,126.02	2,022.12	2.74	-1.84	5.14	2,044.69	274.94	-0.72	3.98	1,636.00
19	50	5	5	**1,229.16**	-	2.71	1,342.20	1,188.32	0.30	-3.44	12.95	1,212.52	13.19	-1.37	10.69	1,037.60
20	150	3	5	2,888.41	-	150.33	3,902.66	3,579.20	31.13	19.30	9.04	**3,676.89**	5,001.70‡	21.44	6.14	2,492.79
21	150	4	3	**1,388.89**	1,915.90	0.00	1,425.62	1,365.13	1.59	-1.74	4.43	1,384.16	117.55	-0.34	3.00	1,113.82
22	150	5	3	**1,446.07**	-	0.50	1,508.76	1,421.57	0.92	-1.72	6.13	1,435.62	113.91	-0.73	5.09	1,040.36
23	150	4	5	2,334.46	-	130.72	3,424.50	3,096.22	5.89	24.60	10.60	**3,116.20**	1,609.56	25.09	9.89	1,671.40
24	150	5	5	1,531.05	-	181.38	1,945.27	1,699.07	2.24	9.89	14.49	**1,744.20**	2,104.79	12.22	11.53	1,382.86

* Hyphen (-) indicates the trigger of the time limit of 3600 s.

† Duality gap (MIPGap) computed internally by the Gurobi optimizer; the default optimality tolerance is 0.01 %.

‡ Trigger of the time limit during the first-stage subproblem.

§ Gurobi optimizer runs out of memory and terminates (Gurobi error 10001: Out of memory).

4.4.2 Case Study: Impacts of Stages and Solution Methods on System Observational Throughput

In this section, we conduct a case study to analyze the impact of having multiple stages on the system observation throughput. To do so, we vary $N \in \{1, 2, 3, 4, 5, 6\}$ on an identical problem with fixed parameters.

Case Study Setup

Following are the fixed parameters:

- **Mission.** Mission planning horizon is referenced to the J2000 epoch, and the duration is 5 days. We set $t_{\text{step}} = 100\,\text{s}$ and $T = 4320$.
- **Satellites.** A fleet of 5 heterogeneous satellites with different orbits (but all are circular) and fuel availabilities. Each satellite has 50 candidate orbital slots that are uniformly distributed within the orbital plane. Table 4.2 shows the key specifications of satellites.

Table 4.2: Key satellite specification parameters.

k	Altitude, km	Inclination, deg.	RAAN, deg.	$c^k_{\max}$, m/s
1	926.16	85.02	196.82	687.40
2	787.89	71.28	159.59	587.85
3	724.69	78.64	12.98	796.70
4	846.24	77.28	296.24	696.53
5	733.40	73.04	98.39	701.13

- **Targets.** A set of 10 spot targets randomly distributed between the latitude interval $[-80°, 80°]$ and the longitude interval $[-180°, 180°]$. Figure 4.4 visualizes the coordinates of the targets. The rewards are target-dependent. Each target p has the time-dependent reward π_p where we assume $\pi_{tp} \in \{0, 1\}$; we randomly generate a random number of blocks of contiguous ones. Figure 4.5 visualizes the observation rewards imposed on each target. Also, we let $\varepsilon_{\min} = 5°$ and $r_{tp} = 1, \forall t \in \mathcal{T}, p \in \mathcal{P}$.

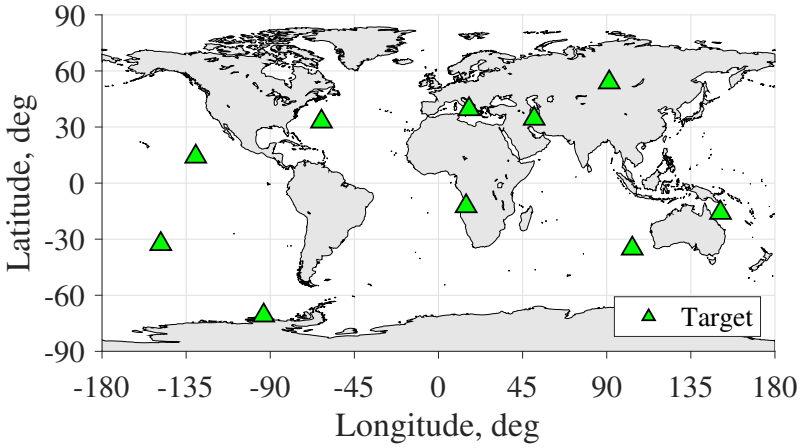


Figure 4.4: Coordinates of the randomly generated targets.

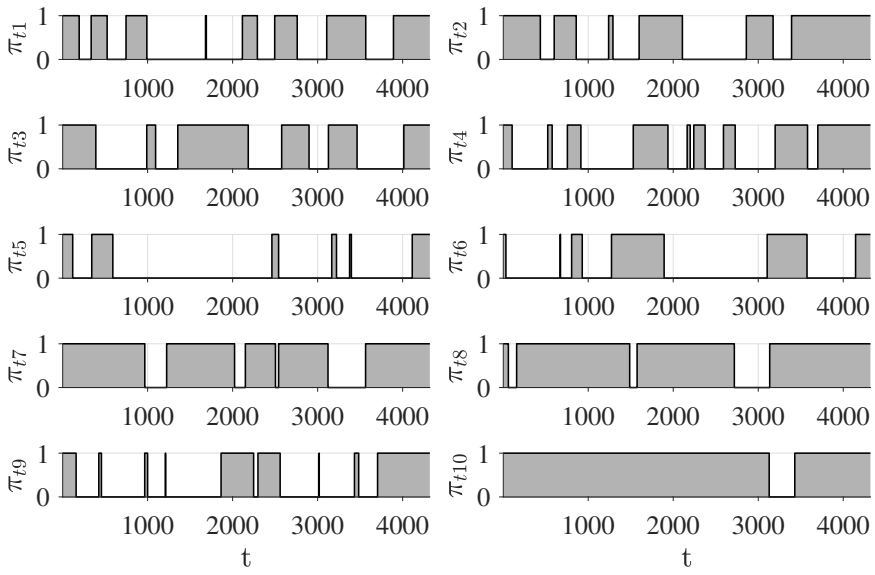


Figure 4.5: Observation rewards set on spot targets. The gray shaded areas refer to ones.

Numerical Results

All instances are solved by MCRP, MPH (for $s \geq 1$), and RHP (for $s \geq 2$), and the Gurobi optimizer solves MCRP and the subproblems of MPH and RHP. We set the runtime limit of 3600 s. Figure 4.6 reports the results of the total observation rewards obtained by all methods per stage. The bottom dashed line shows the reference case without any reconfiguration, which has the score of 4054. By employing reconfiguration, even with only a single stage, the system enhances the observational throughput.

The results conform with our intuitive expectation that, in general, increasing the number of stages increases the total observation rewards by allowing more degrees of freedom for flexibility. (This is not always true because we have resource availability that complicates the problem of multi-stage reconfiguration.) However, in a case with $s = 3$, both RHP and MPH outperform MCRP with the scores of $z_{\text{rhp}} = 4293$, $z_{\text{mph}} = 4281$, and $z = 4278$, respectively. This result is primarily due to the fact that the Gurobi optimizer terminates MCRP early because it triggers the runtime limit of 3600 s and returns the best feasible incumbent solution found thus far. Furthermore, this result implies that unless we solve MCRP to optimality (or with more runtime limit), the heuristic methods can achieve better solutions. To validate this remark, we test again case $s = 3$ of MCRP with a longer runtime limit of 7200 s. In this setting, we obtain $z = 4302$ with the duality gap of 2.09 % that triggers the new runtime limit (hence the solution is still suboptimal); however, the obtained value is greater than the previous one with the runtime limit of 3600 s and those of the heuristic methods. The worst underperformances relative to MCRP are at most 3.62 % for MPH (case $s = 5$) and 2.26 % for RHP (case $s = 5$). These values attest to the high-quality solutions of the heuristic methods.

We also report the computational runtimes for each instance.[2] We observe that MCRP triggers the runtime limit for $s \geq 2$ cases owing to the problem scales while no heuristic

[2]In this analysis, we exclude results of $s = 1$ for MPH and $s = 2$ for RHP because they are identical to MCRP.

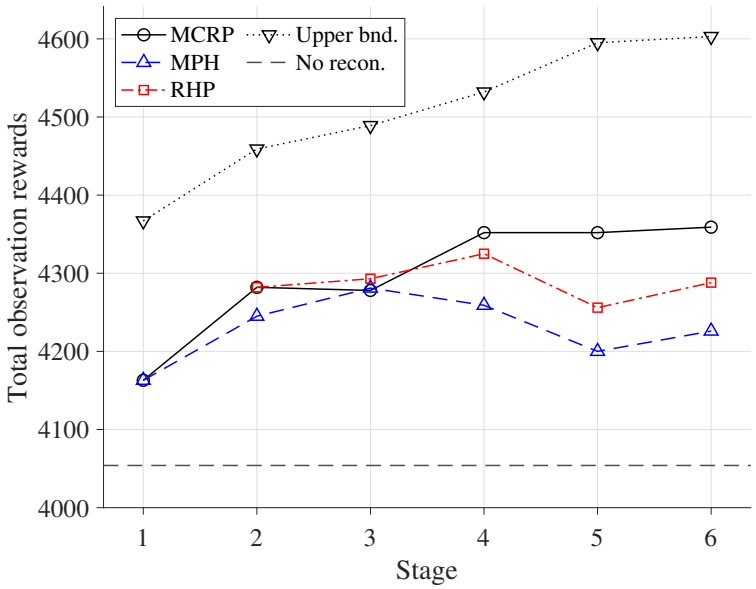


Figure 4.6: Total observation rewards by varying the number of stages.

methods trigger the runtime limit. MPH maintains its runtime less than 14 s (case $s = 2$) for all $s \geq 2$ cases, which is significantly faster than MCRP.

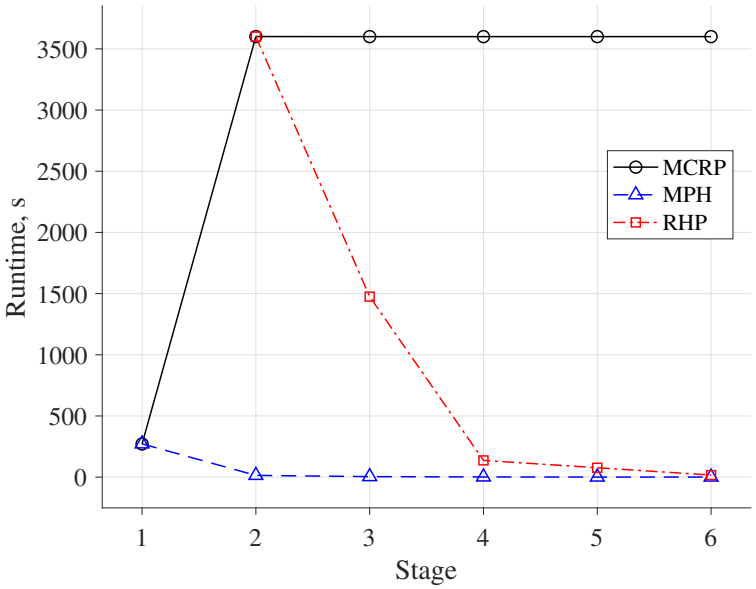


Figure 4.7: Computational runtimes by varying the number of stages.

4.5 Conclusions

This chapter proposes a novel mathematical model to solve the problem of reconfiguring a fleet of satellites to maximize the observation rewards obtained by covering a set of targets over the mission planning horizon while satisfying the individual resource availability constraints. To model stage transitions and the fuel consumption by satellites, we adopt the concept of time-expanded graphs by expanding the nodes (the orbital slots) forward in time and constructing weighted arcs between the nodes of any two adjacent stages. Based on this model, we propose an ILP formulation, which enables the use of commercial MILP solvers for convenience-handling and provably-optimal solutions.

To address the issue of the computational intractability in solving large-scale MCRP, we propose the use of two sequential decision-making approaches: the myopic policy and the rolling horizon procedure. Through the computational experiments, we empirically show that both MPH and RHP provide high-quality solutions in a reasonable computational runtime for instances with uniform observation rewards. In the case of spatiotemporally-varying observation rewards and dynamically-generated targets, RHP outperforms MPH by making informed decisions exploiting the deterministic forecast of the impact of current-stage decisions on an immediate subsequent stage. However, there is no guaranteed outperformance of RHP relative to MPH as the remaining periods are not fully taken into account in the decision-making. We show in the case study that having more stages can increase the total observation rewards, but the quality of actual solutions depends on the performance of an algorithm.

There are two fruitful directions for future research. The first is to improve the fidelity and the applicability of the model. This chapter only considers an ideal case with orbital transfers as decision variables (i.e., the flows). To accurately assess the impact of multi-stage reconfiguration on the scheduling of Earth observation systems, the proposed problem ought to be integrated with a scheduler with various satellite tasks and opera-

tional constraints. Such integration would require modeling efforts in the mapping of complex interactions between various tasks and requirements. The second is to challenge the curse of dimensionality. As shown previously, SP(s) of MPH is RCRP-IRC of Ref. [87], which itself is a combinatorial optimization problem that suffers from the explosion of a solution space in largely-sized instances. Several algorithmic efforts such as asymptotic analysis, node aggregation, approximate dynamic programming, and decomposition-based techniques can reduce the time complexity.

CHAPTER 5
CONCLUSION

The contributions of this dissertation are summarized below.

1. Constellation design for persistent complex regional coverage (Chapter 2)

    - The Access-Pattern-Coverage (APC) decomposition model is developed based on the assumptions of repeating ground track orbits and common ground track constellations motivated by the need to provide persistent regional coverage. The findings show that the relationship between a constellation configuration and its coverage state with respect to a ground target exhibits a circular convolution phenomenon. Using the property of circulant matrices, the APC decomposition model is shown to possess linear properties.
    - Based on the configuration-coverage model, a binary integer linear programming-based constellation design problem is formulated to design a minimum-satellite constellation pattern that satisfies the complex spatiotemporally-varying coverage requirements. The optimized constellation pattern features heterogeneity in satellite specifications and common orbital characteristics and asymmetry in satellite distributions. One key practical implication of this research is that for non-geostationary systems, an optimal constellation pattern dedicated to regional coverage would be an asymmetric constellation in contrast to the symmetric constellation for global coverage.

2. Constellation reconfiguration for adaptive mission planning (Chapter 3)

    - The constellation design problem introduced in Chapter 2, which is in the form of a set covering problem, is not suitable to model the constellation design

aspect in the satellite constellation reconfiguration context. In response to this challenge, this research introduces a new constellation design problem, called the maximum coverage problem (MCP) that enforces the cardinality constraint.

- An integer linear programming formulation of the reconfiguration problem is proposed that integrates the constellation design problem (MCP) and the constellation transfer problem (AP) which are otherwise considered independent and serial in the state-of-the-art. The integrated model enables the exploration of larger design space and a trade-off analysis between the transportation cost and the coverage performance. A variant problem is also introduced that enforces individual resource constraints.

- A dedicated computationally-efficient Lagrangian relaxation-based heuristic method is developed that leverages the assignment problem structure embedded in the problem. Computational experiments attest to the near-optimality of the Lagrangian heuristic solutions and significant improvement in the computational runtime compared to a commercial mixed-integer linear programming solver.

- The developed formulation and the solution method are not restricted to the domain of satellite constellation reconfiguration. For example, the proposed model may be adopted to model the problem of locating a set of first responders (e.g., fire trucks) that maximizes the weighted coverage of demand nodes (e.g., wildfires) while minimizing the cost of deploying first responders from a set of initial locations (e.g, fire stations).

3. Multi-stage constellation reconfiguration problem (Chapter 4)

    - A novel integer linear programming formulation of the multi-stage constellation reconfiguration problem is proposed. This modeling allows us to better understand the hidden design space that is otherwise overlooked with zero or single

reconfiguration stage. The proposed model aims to concurrently optimize the process of reconfigurations over the specified mission planning horizon, thereby incorporating the cost consideration aspect in the long mission horizon.

- Two heuristic solution methods based on the concepts of myopic policy and the rolling horizon procedure are developed to address the issue of computational intractability in solving large-scale problems. This research empirically shows that the myopic policy heuristic can be beneficial for instances with uniform observation rewards, and the rolling horizon procedure can be efficient for instances with dynamic environments owing to the lookahead policy.

Appendices

APPENDIX A

CHAPTER 2 APPENDIX

This chapter lists all appendices for Chapter 2.

A.1 Expanded Ground Track View

The *expanded ground track view* spatially expands an ordinary ground track of a satellite and visualizes its ground track relative to the area of interest throughout the simulation period T_{sim}. The area of interest and its mirrored images are positioned throughout the plot (the red squares in Figure A.1) to provide spatial references. The expanded ground track view is especially useful when visualizing and correlating the access profile and the actual satellite ground track.

The following properties of the expanded ground track view are formalized for the repeating ground track with the period ratio of $\tau = N_{\text{P}}/N_{\text{D}}$.

1. The magnitude of the longitudinal angular displacement of the expanded ground track is $360|N_{\text{P}} - N_{\text{D}}|$ degrees for prograde orbits or $360(N_{\text{P}} + N_{\text{D}})$ degrees for retrograde orbits [90]. Here, the longitudinal angular displacement of the expanded ground track is defined as the total angular displacement required to repeat the ground track, measured along the axis of longitude in the direction of the satellite's motion.

2. The mirrored images of the area of interest are separated by 360 degrees.

A.2 Derivation of the Coverage Timeline

To prove the circular convolution phenomenon, show that Eq. (2.14) is identical to Eq. (2.20). Begin by expanding Eq. (2.14), which is the summation of all access profiles:

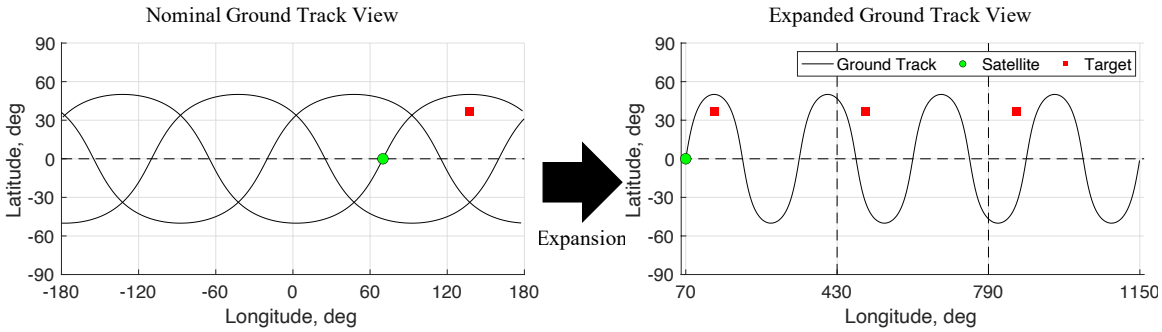


Figure A.1: Full expansion of a ground track of $\mathbf{\text{œ}}_0 = [4/1, 0, 50°, 0°, 350.2°, 0°]$ (J2000).

$$b_j[n] = v_{1,j}[n] + v_{2,j}[n] + \cdots + v_{N,j}[n] \tag{A.1}$$

Each term of Eq. (A.1) can be represented as a multiple of $v_{0,j}[n]$ and permutation matrix $\boldsymbol{P}_\pi^{n_k}$ due to the cyclic property of the assumed formulation. Recalling the definition from Eq. (2.12):

$$v_{k,j}[n] = \boldsymbol{P}_\pi^{n_k} v_{0,j}[n]$$

where $\boldsymbol{P}_\pi$ is a permutation matrix with the dimension $(L \times L)$ shown below. Note that $\boldsymbol{I} = \boldsymbol{P}_\pi^0 = \boldsymbol{P}_\pi^L$.

$$\boldsymbol{P}_\pi = \begin{bmatrix} 0 & 0 & 0 & \cdots & 1 \\ 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}$$

Substituting Eq. (2.12) into Eq. (A.1), we get the following equation:

$$b_j[n] = (\boldsymbol{P}_\pi^{n_1} + \boldsymbol{P}_\pi^{n_2} + \cdots + \boldsymbol{P}_\pi^{n_N}) v_{0,j}[n] \tag{A.2}$$

Eq. (A.2) is a superposition of cyclically shifted access profiles referenced to a seed satellite access profile. Here, n_k denotes the index of the relative time shift of the kth access

profile with respect to the seed satellite access profile. Instead of only indicating the indices where only access profiles exist, one can generalize this to all time steps $n \in \{0, ..., L-1\}$ following the definition of the constellation pattern vector in Eq. (2.17). Hence, Eq. (A.2) can be further deduced as:

$$b_j[n] = \left(x[0]\boldsymbol{P}_\pi^0 + x[1]\boldsymbol{P}_\pi^1 + \cdots + x[L-1]\boldsymbol{P}_\pi^{L-1}\right)v_{0,j}[n] \tag{A.3}$$

The terms within parentheses in Eq. (A.3) is identical to the alternative analytical definition of the circulant matrix:

$$\boldsymbol{X} \triangleq x[0]\boldsymbol{I} + x[1]\boldsymbol{P}_\pi^1 + \cdots + x[L-1]\boldsymbol{P}_\pi^{L-1} \tag{A.4}$$

Finally, substituting Eq. (A.4) into Eq. (A.3), we get:

$$\boldsymbol{b}_j = \boldsymbol{X}\boldsymbol{v}_{0,j} \tag{A.5}$$

Using the commutative property of the circular convolution operator, Eq. (A.5):

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

where

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

as defined in Eq. (2.21).

This is identical to the definition of the circular convolution in Eq. (2.20), thereby proving the circular convolutional nature of the formulation under the aforementioned assumptions.

A.3 Integrating the Developed Method into Constellation Design Process

This appendix introduces an example approach to integrate the developed method into the satellite constellation design process. As discussed earlier, the developed satellite constellation pattern design method needs the seed satellite orbital elements as its input. In this appendix, we introduce an approach to efficiently integrate the determination of the seed satellite orbital elements $\mathbf{oe}_0$ and the determination of the constellation pattern $\boldsymbol{x}$ (i.e., the developed method).

First, note that although $\mathbf{oe}_0$ contains six orbital elements $(\tau, e, i, \omega, \Omega_0, M_0)$, we only have five degrees of freedom. The initial mean anomaly of the seed satellite M_0 can be set to zero without loss of generality. This is because, as shown in Eq. (2.37), Ω_0 and n_k can be chosen such that any solution with an arbitrarily chosen M_0 can be converted into an equivalent solution with $M_0 = 0°$[1].

The design space of the remaining five orbital elements can be narrowed down even further by considering the launch and mission requirements. As an example, we consider the case used in Example 2 in Section 2.6 and provide a walk-through process.

1. Suppose there is demand for increased communications capacity (i.e., increased satellite diversity) during a particular time interval of a day that repeats daily (e.g., Internet rush hour) over Atlanta, Georgia ($\{(\phi = 34.75°\text{N}, \lambda = 84.39°\text{W})\}$). Translating this demand, the time-varying coverage requirement $\boldsymbol{f}$ is derived (see Example 2). The communications quality-of-service requirement further enforces consistency in data round-trip latency throughout the mission duration; hence, a circular orbit is desired. The period ratio and the minimum elevation angle are assumed to be derived a priori based on mission-related requirements: $\tau = 12/1$ and $\varepsilon_{\min} = 5°$.

2. Based on the set of mission requirements and parameters ($T_\text{r} = 86\,400\,\text{s}$, $e = 0$, and $\tau = 12/1$), the inclination of the orbit is readily derived, which is approximately

[1] Strictly speaking, there are only a finite number of possible discrete values for M_0 due to the discretization used in this problem.

102.9°. Note that since the repeat period T_r is exactly given together with τ and e, there is no degree of freedom for trading off the altitude and the inclination. In this case, since the repeat period is exactly 86 400 s, the orbit needs to be a repeating sun-synchronous orbit.

3. At this point, the only leftover variable is Ω_0, which dictates the shift of the common ground track along the longitudinal direction. The RAAN of the seed satellite Ω_0 can be determined either by an analytical heuristic method or by a numerical optimization.

    a) An analytical heuristic approach can determine Ω_0 such that the common ground track is symmetric about the longitude of the target point (see Figure A.2). Solving for the corresponding RAAN value yields $\Omega_0 = 98.3°$. Note that another symmetry exists further offsetting Ω_0 value.

    b) A single-variable optimization can be performed to determine the value of Ω_0. Ideally, we prefer to use the number of satellites as the metric, but this cannot be evaluated without $\boldsymbol{x}$. Instead, an effective metric can be the coverage over the area of interest. Note that the values of $\boldsymbol{\text{œ}}_0$ maximizing the coverage does not necessarily lead to a minimum number of satellites, but as shown later, it is a good approximation to use.

4. Using the obtained seed satellite orbital elements, the optimization of the constellation pattern vector can be performed following the APC-based methods developed in Chapter 2.

As we evaluate the efficiency of the developed integrated heuristics and BILP methods, we compare them against a more straightforward approach, where both $\boldsymbol{\text{œ}}_0$ and $\boldsymbol{x}$ are optimized as variables simultaneously against the objective function of the number of satellites. In fact, this formulation is the most direct representation of our goal; however, since it is a

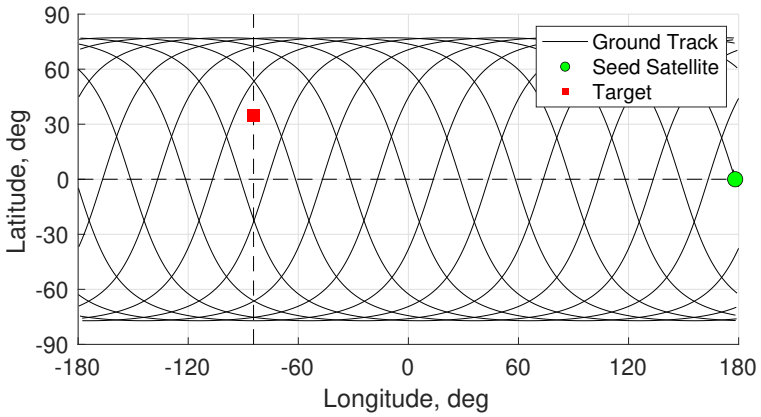


Figure A.2: Alignment of the ground track such that it is symmetric about the longitude of the target.

mixed-integer nonlinear optimization problem, we cannot leverage the developed method and therefore can only use generic solvers (e.g., genetic algorithm). Here, we aim to show that, by incorporating the developed method into this process, we can achieve a much better performance than this classical integrated method.

In Table A.1, Method 1 refers to the heuristics approach that finds Ω_0 using symmetry, which is the actual method used in Example 2. Method 2 refers to the two-stage optimization where the first stage is the meta-heuristic optimization of Ω_0 and the second stage is the BILP optimization of $\boldsymbol{x}$. Lastly, Method 3 is the simultaneous optimization of both Ω_0 and $\boldsymbol{x}$ via meta-heuristic optimization. For Methods 2 and 3, a genetic algorithm by MATLAB is used with the default settings.

Table A.1: Comparison of different methods for integrated optimization.

Method	Number of Satellites	Computational Time, s
1	24	3712.0
2	25	Stage 1: 477.1 Stage 2: 2086.0
3	75	2482.1 (Population: 100)
	36	5854.7 (Population: 200)
	32	10635.1 (Population: 300)

The results show that both Methods 1 and 2 are effective in finding the optimal solution; the only difference in these two methods is in the optimization of Ω_0. On the other hand,

Method 3 requires longer computational time, while only showing poor results. These results demonstrate the utility of the developed method when integrated into the satellite constellation design process.

A.4 Derivation of the RAAN phasing

This appendix derives

$$\Omega_k = n_k \frac{2\pi N_\mathrm{D}}{L} + \Omega_0$$

in Eq. (2.37b). Define $\Delta\Omega = (\Omega_k - \Omega_0)/n_k$. Our goal is to prove $\Delta\Omega = 2\pi N_\mathrm{D}/L$.

This expression comes from Figure 2.5. In order to achieve a constellation that separates away from each other by t_step over a common ground track, $\Delta\Omega$ needs to be defined as the difference between Earth's rotation and the angular displacement due to the RAAN precession during a time interval $[0, t_\mathrm{step}]$. More specifically,

$$\Delta\Omega = (\omega_\oplus - \dot{\Omega}) t_\mathrm{step} \tag{A.6}$$

Since $t_\mathrm{step} = T_\mathrm{r}/L$, substituting in Eq. (2.1) yields $t_\mathrm{step} = N_\mathrm{D} T_\mathrm{G}/L$. Plugging this into Eq. (A.6), we get:

$$\Delta\Omega = (\omega_\oplus - \dot{\Omega}) \frac{N_\mathrm{D} T_\mathrm{G}}{L}$$

Since, $T_\mathrm{G} = 2\pi/(\omega_\oplus - \dot{\Omega})$ (Eq. (2.2b)), we get:

$$\Delta\Omega = \frac{2\pi N_\mathrm{D}}{L}$$

REFERENCES

[1] *Indian regional navigation satellite system (irnss)*, https://www.isro.gov.in/irnss-programme, Accessed January 15, 2019,

[2] *Quasi-zenith satellite system (qzss)*, http://qzss.go.jp/en/, Accessed January 15, 2019,

[3] D. Diekelman, "Design guidelines for post-2000 constellations," in *Mission Design & Implementation of Satellite Constellations*, J. C. van der Ha, Ed., Dordrecht: Springer Netherlands, 1998, pp. 11–21, ISBN: 978-94-011-5088-0.

[4] H. 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.

[5] E. Lutz, M. Werner, and A. Jahn, *Satellite systems for personal and broadband communications*. Springer Science & Business Media, 2012.

[6] J. G. Walker, "Circular orbit patterns providing continuous whole earth coverage," Royal Aircraft Establishment Farnborough (United Kingdom), Tech. Rep., 1970.

[7] ——, "Continuous whole-earth coverage by circular-orbit satellite patterns," Royal Aircraft Establishment Farnborough (United Kingdom), Tech. Rep., 1977.

[8] ——, "Satellite constellations," *Journal of the British Interplanetary Society*, vol. 37, pp. 559–572, 1984.

[9] R. D. Lüders, "Satellite networks for continuous zonal coverage," *ARS Journal*, vol. 31, no. 2, pp. 179–184, 1961. eprint: https://doi.org/10.2514/8.5422.

[10] R. Lüders and L. Ginsberg, "Continuous zonal coverage-a generalized analysis," in *Mechanics and Control of Flight Conference*, 1974, p. 842. eprint: https://arc.aiaa.org/doi/pdf/10.2514/6.1974-842.

[11] D. C. Beste, "Design of satellite constellations for optimal continuous coverage," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-14, no. 3, pp. 466–473, 1978.

[12] 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.

[13] A. H. Ballard, "Rosette constellations of earth satellites," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-16, no. 5, pp. 656–673, 1980.

[14] J. E. Draim, "A common-period four-satellite continuous global coverage constellation," *Journal of Guidance, Control, and Dynamics*, vol. 10, no. 5, pp. 492–499, 1987. eprint: https://doi.org/10.2514/3.20244.

[15] J. R. Wertz, *Mission Geometry: Orbit and Constellation Design and Management: Spacecraft Orbit and Attitude Systems*, ser. Space technology library. Microcosm Press, 2001.

[16] J. M. Hanson, M. J. Evans, and R. E. Turner, "Designing good partial coverage satellite constellations," *Journal of the Astronautical Sciences*, vol. 40, no. 2, pp. 215–239, 1992.

[17] D.-M. Ma and W.-C. Hsu, "Exact design of partial coverage satellite constellations over oblate earth," *Journal of Spacecraft and Rockets*, vol. 34, no. 1, pp. 29–35, 1997. eprint: https://doi.org/10.2514/2.3188.

[18] M. Pontani and P. Teofilatto, "Satellite constellations for continuous and early warning observation: A correlation-based approach," *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 4, pp. 910–921, 2007. eprint: https://doi.org/10.2514/1.23094.

[19] W. A. Crossley and E. A. Williams, "Simulated annealing and genetic algorithm approaches for discontinuous coverage satellite constellation design," *Engineering Optimization*, vol. 32, no. 3, pp. 353–371, 2000. eprint: https://doi.org/10.1080/03052150008941304.

[20] Y. Ulybyshev, "Satellite constellation design for complex coverage," *Journal of Spacecraft and Rockets*, vol. 45, no. 4, pp. 843–849, 2008. eprint: https://doi.org/10.2514/1.35369.

[21] G. Dutruel-Lecohier and M. B. Mora, "Orion — a constellation mission analysis tool," in *Mission Design & Implementation of Satellite Constellations*, J. C. van der Ha, Ed., Dordrecht: Springer Netherlands, 1998, pp. 373–393, ISBN: 978-94-011-5088-0.

[22] Y. Ulybyshev, "Satellite constellation design for continuous coverage: Short historical survey, current status and new solutions," *Proceedings of Moscow Aviation Institute*, vol. 13, no. 34, pp. 1–25, 2009.

[23] D. Mortari, M. P. Wilkins, and C. Bruccoleri, "The flower constellations," *Journal of the Astronautical Sciences*, vol. 52, no. 1, pp. 107–127, 2004.

[24] D. Mortari and M. P. Wilkins, "Flower constellation set theory. part i: Compatibility and phasing," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 44, no. 3, pp. 953–962, 2008.

[25] M. P. Wilkins and D. Mortari, "Flower constellation set theory part ii: Secondary paths and equivalency," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 44, no. 3, pp. 964–976, 2008.

[26] H. Lee, K. Ho, S. Shimizu, and S. Yoshikawa, "A semi-analytical approach to satellite constellation design for regional coverage," in *AAS/AIAA Astrodynamics Specialist Conference, 2018*, P. Singla, R. Weisman, B. Marchand, and B. Jones, Eds., ser. Advances in the Astronautical Sciences, Univelt Inc., Jan. 2018, pp. 171–190, ISBN: 9780877036579.

[27] S. Vtipil and B. Newman, "Determining an earth observation repeat ground track orbit for an optimization methodology," *Journal of Spacecraft and Rockets*, vol. 49, no. 1, pp. 157–164, 2012. eprint: https://doi.org/10.2514/1.A32038.

[28] C. Bruccoleri, "Flower constellation optimization and implementation," Ph.D. dissertation, Texas AM University, 2007.

[29] T. Wu, S. Wu, and L. Zhu, "Design of common track satellite constellations for optimal regional coverage," in *2006 6th International Conference on ITS Telecommunications*, IEEE, 2006, pp. 1252–1255.

[30] 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, Aug. 2013.

[31] M. A. Chylla and C. D. Eagle, "Efficient computation of satellite visibility periods," in *Spaceflight Mechanics 1992*, 1992, pp. 823–834.

[32] S. Alfano, J. Negron David, and J. L. Moore, "Rapid determination of satellite visibility periods," *Journal of the Astronautical Sciences*, vol. 40, no. 2, pp. 281–296, Jun. 1992.

[33] C. Han, X. Gao, and X. Sun, "Rapid satellite-to-site visibility determination based on self-adaptive interpolation technique," *Science China Technological Sciences*, vol. 60, no. 2, pp. 264–270, 2017.

[34] *Systems tool kit help guide*, http://help.agi.com/stk/, Accessed April 28, 2019,

[35] R. M. Gray, "Toeplitz and circulant matrices: A review," *Foundations and Trends® in Communications and Information Theory*, vol. 2, no. 3, pp. 155–239, 2006.

[36] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*, 3rd. USA: Prentice Hall Press, 2009, ISBN: 0131988425.

[37] Gurobi Optimization, Inc., *Gurobi optimizer reference manual*, 2016.

[38] S. Lee, Y. Wu, and D. Mortari, "Satellite constellation design for telecommunication in antarctica," *International Journal of Satellite Communications and Networking*, vol. 34, no. 6, pp. 725–737, 2016. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/sat.1128.

[39] P. Wessel and W. H. F. Smith, *Shoreline boundary between antarctic grounding line and the ocean, 2014 (full-resolution)*, National Geospatial-Intelligence Agency, 2014.

[40] J. Q. Chambers *et al.*, "Regional ecosystem structure and function: Ecological insights from remote sensing of tropical forests," *Trends in Ecology & Evolution*, vol. 22, no. 8, pp. 414–423, 2007.

[41] T. Rientjes, A. T. Haile, and A. A. Fenta, "Diurnal rainfall variability over the upper blue nile basin: A remote sensing based approach," *International Journal of Applied Earth Observation and Geoinformation*, vol. 21, pp. 311–325, 2013.

[42] *Major river basins of the world*, https://datacatalog.worldbank.org/dataset/major-river-basins-world, Accessed January 11, 2020,

[43] X. He, H. Li, L. Yang, and J. Zhao, "Reconfigurable satellite constellation design for disaster monitoring using physical programming," *International Journal of Aerospace Engineering*, vol. 2020, H. Asadov, Ed., p. 8 813 685, 2020.

[44] Y. Chen, V. Mahalec, Y. Chen, X. Liu, R. He, and K. Sun, "Reconfiguration of satellite orbit for cooperative observation using variable-size multi-objective differential evolution," *European Journal of Operational Research*, vol. 242, no. 1, pp. 10–20, 2015.

[45] O. L. de Weck, R. de Neufville, and M. Chaize, "Staged deployment of communications satellite constellations in low earth orbit," *Journal of Aerospace Computing, Information, and Communication*, vol. 1, no. 3, pp. 119–136, 2004. eprint: https://doi.org/10.2514/1.6346.

[46] D. Arnas and R. Linares, "Uniform satellite constellation reconfiguration," *Journal of Guidance, Control, and Dynamics*, vol. 45, no. 7, pp. 1241–1254, 2022. eprint: https://doi.org/10.2514/1.G006514.

[47] M. P. Ferringer, D. B. Spencer, and P. Reed, "Many-objective reconfiguration of operational satellite constellations with the large-cluster epsilon non-dominated sorting

genetic algorithm-ii," in *2009 IEEE Congress on Evolutionary Computation*, 2009, pp. 340–349.

[48] J. J. Davis, "Constellation reconfiguration: Tools and analysis," Ph.D. dissertation, Texas A&M University, 2010.

[49] M. Fakoor, M. Bakhtiari, and M. Soleymani, "Optimal design of the satellite constellation arrangement reconfiguration process," *Advances in Space Research*, vol. 58, no. 3, pp. 372–386, 2016.

[50] G. Denis, H. de Boissezon, S. Hosford, X. Pasco, B. Montfort, and F. Ranera, "The evolution of earth observation satellites in europe and its impact on the performance of emergency response services," *Acta Astronautica*, vol. 127, pp. 619–633, 2016.

[51] S. Voigt *et al.*, "Global trends in satellite-based emergency mapping," *Science*, vol. 353, no. 6296, pp. 247–252, 2016. eprint: https://www.science.org/doi/pdf/10.1126/science.aad8728.

[52] X. Wang, G. Wu, L. Xing, and W. Pedrycz, "Agile earth observation satellite scheduling over 20 years: Formulations, methods, and future directions," *IEEE Systems Journal*, vol. 15, no. 3, pp. 3881–3892, 2021.

[53] S. W. Paek, S. Kim, and O. de Weck, "Optimization of reconfigurable satellite constellations using simulated annealing and genetic algorithm," *Sensors*, vol. 19, no. 4, 2019.

[54] C. N. McGrath and M. Macdonald, "General perturbation method for satellite constellation reconfiguration using low-thrust maneuvers," *Journal of Guidance, Control, and Dynamics*, vol. 42, no. 8, pp. 1676–1692, 2019. eprint: https://doi.org/10.2514/1.G003739.

[55] Z. Zhang, N. Zhang, Y. Jiao, H. Baoyin, and J. Li, "Multi-tree search for multi-satellite responsiveness scheduling considering orbital maneuvering," *IEEE Transactions on Aerospace and Electronic Systems*, pp. 1–1, 2021.

[56] L. Appel, M. Guelman, and D. Mishne, "Optimization of satellite constellation reconfiguration maneuvers," *Acta Astronautica*, vol. 99, pp. 166–174, 2014.

[57] R. S. Legge Jr, "Optimization and valuation of recongurable satellite constellations under uncertainty," Ph.D. dissertation, Massachusetts Institute of Technology, 2014.

[58] O. L. de Weck, U. Scialom, and A. Siddiqi, "Optimal reconfiguration of satellite constellations with the auction algorithm," *Acta Astronautica*, vol. 62, no. 2, pp. 112–130, 2008.

[59] H. 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. eprint: https://doi.org/10.2514/1.A34657.

[60] K.-J. Zhu, J.-F. Li, and H.-X. Baoyin, "Satellite scheduling considering maximum observation coverage time and minimum orbital transfer fuel cost," *Acta Astronautica*, vol. 66, no. 1, pp. 220–229, 2010.

[61] J. J. Bartholdi, J. B. Orlin, and H. D. Ratliff, "Cyclic scheduling via integer programs with circular ones," *Operations Research*, vol. 28, no. 5, pp. 1074–1085, 1980. eprint: https://doi.org/10.1287/opre.28.5.1074.

[62] J. J. Bartholdi, "A guaranteed-accuracy round-off algorithm for cyclic scheduling and set covering," *Operations Research*, vol. 29, no. 3, pp. 501–510, 1981. eprint: https://doi.org/10.1287/opre.29.3.501.

[63] M. Avendaño and D. Mortari, "A closed-form solution to the minimum ΔV_{tot}^2 Lambert's problem," *Celestial Mechanics and Dynamical Astronomy*, vol. 106, no. 1, pp. 25–37, 2009.

[64] A. J. Hoffman and J. B. Kruskal, "Integral boundary points of convex polyhedra," in *Linear Inequalities and Related Systems. (AM-38)*. Princeton University Press, 1956, vol. 38, pp. 223–246.

[65] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[66] D. P. Bertsekas, "A new algorithm for the assignment problem," *Mathematical Programming*, vol. 21, no. 1, pp. 152–171, 1981.

[67] D. P. Bertsekas and J. Eckstein, "Dual coordinate step methods for linear network flow problems," *Mathematical Programming*, vol. 42, no. 1, pp. 203–243, 1988.

[68] D. Vallado, *Fundamentals of Astrodynamics and Applications*, ser. Space technology library. Microcosm Press, 2013, ISBN: 9781881883180.

[69] H. Lee and K. Ho, "Binary integer linear programming formulation for optimal satellite constellation reconfiguration," in *AAS/AIAA Astrodynamics Specialist Conference*, Aug. 2020.

[70] C. ReVelle, M. Scholssberg, and J. Williams, "Solving the maximal covering location problem with heuristic concentration," *Computers & Operations Research*, vol. 35, no. 2, pp. 427–435, 2008, Part Special Issue: Location Modeling Dedicated to the memory of Charles S. ReVelle.

[71] N. Megiddo, E. Zemel, and S. L. Hakimi, "The maximum coverage location problem," *SIAM Journal on Algebraic Discrete Methods*, vol. 4, no. 2, pp. 253–261, 1983.

[72] R. Church and C. ReVelle, "The maximal covering location problem," *Papers in Regional Science*, vol. 32, no. 1, pp. 101–118, 1974.

[73] H. Y. Yacov, L. S. Lasdon, and D. A. Wismer, "On a bicriterion formulation of the problems of integrated system identification and system optimization," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 1, no. 3, pp. 296–297, 1971.

[74] M. L. Fisher, "The lagrangian relaxation method for solving integer programming problems," *Management Science*, vol. 50, no. 12_supplement, pp. 1861–1871, 2004. eprint: https://doi.org/10.1287/mnsc.1040.0263.

[75] M. Held and R. M. Karp, "The traveling-salesman problem and minimum spanning trees: Part II," *Mathematical Programming*, vol. 1, no. 1, pp. 6–25, 1971.

[76] M. Held, P. Wolfe, and H. P. Crowder, "Validation of subgradient optimization," *Mathematical Programming*, vol. 6, no. 1, pp. 62–88, 1974.

[77] D. Bertsimas and J. Tsitsiklis, *Introduction to linear optimization*. Athena Scientific, 1997.

[78] M. Guignard, "Lagrangian relaxation," in *Encyclopedia of Operations Research and Management Science*, S. I. Gass and M. C. Fu, Eds. Boston, MA: Springer US, 2013, pp. 845–860, ISBN: 978-1-4419-1153-7.

[79] A. M. Geoffrion, "Lagrangean relaxation for integer programming," in *Approaches to Integer Programming*, M. L. Balinski, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1974, pp. 82–114, ISBN: 978-3-642-00740-8.

[80] H. Lee and K. Ho, "A lagrangian relaxation-based heuristic approach to regional constellation reconfiguration problem," in *AAS/AIAA Astrodynamics Specialist Conference*, Aug. 2021.

[81] R. D. Galvão and C. ReVelle, "A lagrangean heuristic for the maximal covering location problem," *European Journal of Operational Research*, vol. 88, no. 1, pp. 114–123, 1996.

[82] A. Golkar and I. Lluch i Cruz, "The federated satellite systems paradigm: Concept and business case evaluation," *Acta Astronautica*, vol. 111, pp. 230–248, 2015.

[83] P. Wang, G. Reinelt, P. Gao, and Y. Tan, "A model, a heuristic and a decision support system to solve the scheduling problem of an earth observing satellite constellation,"

Computers & Industrial Engineering, vol. 61, no. 2, pp. 322–335, 2011, Combinatorial Optimizatiion in Industrial Engineering.

[84] W. J. Wolfe and S. E. Sorensen, "Three scheduling algorithms applied to the earth observing systems domain," *Management Science*, vol. 46, no. 1, pp. 148–166, 2000. eprint: https://doi.org/10.1287/mnsc.46.1.148.15134.

[85] J. Kim and J. Ahn, "Integrated framework for task scheduling and attitude control of multiple agile satellites," *Journal of Aerospace Information Systems*, vol. 18, no. 8, pp. 539–552, 2021. eprint: 10.2514/1.I010910.

[86] T. Sarton du Jonchay, H. Chen, O. Gunasekara, and K. Ho, "Framework for modeling and optimization of on-orbit servicing operations under demand uncertainties," *Journal of Spacecraft and Rockets*, vol. 58, no. 4, pp. 1157–1173, 2021.

[87] H. Lee and K. Ho, *Regional constellation reconfiguration problem: Integer linear programming formulation and lagrangian heuristic method*, 2022.

[88] W. B. Powell, "What you should know about approximate dynamic programming," *Naval Research Logistics (NRL)*, vol. 56, no. 3, pp. 239–249, 2009. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/nav.20347.

[89] S. Sethi and G. Sorger, "A theory of rolling horizon decision making," *Annals of Operations Research*, vol. 29, no. 1, pp. 387–415, 1991.

[90] M. Kim, "Theory of satellite ground-track crossovers," *Journal of Geodesy*, vol. 71, no. 12, pp. 749–767, 1997.

VITA

Hang Woon Lee was born in Seoul, South Korea. Hang Woon received an S.B. degree in aerospace engineering from Massachusetts Institute of Technology (2015) and an M.S. degree in aerospace engineering from the University of Illinois at Urbana-Champaign (2018). He joined the Daniel Guggenheim School of Aerospace Engineering at Georgia Institute of Technology in 2019. He is the recipient of the National Science Foundation Graduate Research Fellowship (2018) and the Molly K. Macauley Award (2020) from the American Astronautical Society. His industry experience includes SpaceX and Planet. His research interests reside at the intersection of space systems engineering, operations research, and astrodynamics.