FRAMEWORK FOR THE DESIGN AND OPERATIONS OF SUSTAINABLE ON-ORBIT SERVICING INFRASTRUCTURES DEDICATED TO GEOSYNCHRONOUS SATELLITES

A Dissertation
Presented to
The Academic Faculty

By

Tristan Sarton du Jonchay

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

December 2022

FRAMEWORK FOR THE DESIGN AND OPERATIONS OF SUSTAINABLE ON-ORBIT SERVICING INFRASTRUCTURES DEDICATED TO GEOSYNCHRONOUS SATELLITES

Thesis committee:

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

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

Dr. Sandra Magnus
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Mark Whorton
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Paul Grogan
School of Systems and Enterprises
Stevens Institute of Technology

Date approved: November 8, 2022

For my mother Béatrice

TABLE OF CONTENTS

ACKNOWLEDGMENTS

What a journey! Seven years after my first interaction with Prof. Ho, and here I am, living the last few moments as a Ph.D. student, remembering all the fun (and less fun) moments that marked my life as a graduate student.

Before all, I could not thank Prof. Ho deeply enough for the opportunity he gave me to work on truly distinctive research projects ranging from human space exploration mission design and space commercialization to on-orbit servicing. His support, enthusiasm, and unique perspective on complex problems helped me all along my Ph.D. studies. Last but not least, what I will remember the most about Prof. Ho is his intentional desire to create a stress-free work environment for his students to thrive and succeed.

I would also like to take this opportunity to extend my thanks to Prof. Lightsey who has done a tremendous job at making the Space Systems Design Lab at Georgia Tech a truly fun and interesting place to be. With an ambitious and exciting vision for space like his, there is no doubt that the Space Systems Design Lab will keep growing and attracting talented students.

I also would like to thank Dr Magnus, Dr Whorton, and Dr Grogan for their intellectual input and guidance as part of my dissertation committee.

My life at UIUC and Georgia Tech would not have been as colorful without the truly enthusiastic, loving, and fun people I met there, all from different countries and cultures: Sulagna, Olivier, Nnamdi, Hang Woon, Hao, Onalli, Kento, Mikael, and the people at the Space Systems Optimization Group.

Finally, I am indebted to my dad and siblings for their unconditional support and belief in me all along this journey. I particularly grateful to my late mother, Béatrice, who keeps supporting me wherever she is.

LIST OF TABLES

LIST OF FIGURES

SUMMARY

After being a concept for decades, the on-orbit servicing industry is finally taking off, with national space agencies and private organizations developing and planning soon-to-be-launched space infrastructures that will revolutionize the way humans operate in space. The advent of this new industry comes at a time when the geosynchronous orbit (GEO) satellite industry faces various pressures whether it be because of ageing fleets or increased competition from nimbler Low Earth Orbit (LEO) and Medium Earth Orbit (MEO) constellations. A new symbiotic relationship is emerging between early OOS players who seek customers and the GEO satellite operators who aim to revive the competitiveness of their fleets.

The first OOS infrastructures will be simple ones, involving a few servicers offering a narrow set of services. These servicers will provide services to a few satellites before running out of propellant and getting discarded in a graveyard orbit or into the atmosphere. However, as technology matures and demand for on-orbit services increases, OOS infrastructures will become more versatile and involve additional elements, such as orbital depots, to enable the sustainable operations of a wide variety of servicers. Thus, planning OOS missions will involve not only finding the best route for every single servicer but also optimizing the in-space supply chain of commodities needed to support the long-term operations of the servicers and their client satellites.

This dissertation presents an OOS planning framework that simultaneously computes the optimal route of the servicers and plans the in-space supply chain of the supporting commodities. The second chapter gives the background of OOS in GEO and the literature review for OOS planning relevant to the work presented in this thesis. The third chapter presents the mission scenario investigated in this work. The fourth chapter generalizes the Time-Expanded Generalized Multi Commodity Network Flow (TE-GMCNF) model used in recent state-of-the-art space logistics studies to accurately model the operations

of the servicers across a network of customer satellites and orbital depots. The Rolling Horizon (RH) approach is adapted to the OOS context to properly model uncertain service demand arising from customer satellites. The fifth chapter generalizes the mathematical formulation at the core of the framework developed in chapter 4 to model all kinds of user-defined trajectories and servicer propulsion technologies, such as high-thrust, low-thrust, and/or multimodal servicers. (Multimodal servicers are defined to be equipped with both high-thrust and low-thrust engines.) An assumption inherent to chapter 4 and chapter 5 is that the nodes of the networks are all co-located along the same orbit. Chapter 6 relaxes this assumption by extending the framework developed in chapter 4 through the computation of the relative dynamics of network nodes distributed across orbits of various shapes and orientations. Thus, chapter 6, unlike chapter 4 and chapter 5, optimizes the operations of OOS infrastructures over a network with time-varying arc costs.

CHAPTER 1
INTRODUCTION

The fleet of GEO satellites represents a fundamental piece of modern infrastructure providing a broad range of indispensable services to human civilization, from communications and television broadcasting to weather forecasting and defense and intelligence applications. However, despite our apparent dependence on this critical infrastructure, GEO satellite operators are facing rising pressures on several fronts. First, the fleet of GEO satellites is ageing. Their days are counted before the need arises to de-orbit and replace them. Second, replacing ageing GEO satellites is a costly and timely process with little attention given to ensuring the sustainability of the orbital environment. GEO satellites are typically designed for long lifespans ranging from 10 to 20 years, with many technological redundancies to ensure a non-zero probability of generating profits until their planned end of life. In addition, ageing satellites must be de-orbited in a graveyard orbit which, although currently of little use for terrestrial applications, may very well end up being of significant importance as activities in cislunar space amplify. Orbital debris represent a serious threat to our future in space; thus, every single source of space junk must be accounted for and mitigated. Finally, the GEO infrastructure faces competition from rising LEO and MEO satellite mega-constellations comprising hundreds of small, relatively cheap satellites that can easily be de-orbited into the atmosphere and replaced. Because of a fast and cheap satellite turnover, mega-constellations can quickly respond to technological changes and receive significant upgrades. Such is not the case of GEO satellites.

The traditional paradigm of long-lived but poorly adaptable GEO satellites may however be shifting towards a more sustainable future with the advent of On-Orbit Servicing (OOS) technologies. OOS technologies comprise a suite of robotic tools, servicing spacecraft (called servicers), resupply vehicles (e.g., propellant tankers), and orbital depots meant to maintain the Earth's orbital environment. In particular, the GEO fleet represents one of OOS's promising early markets. OOS infrastructures will first be providing services such as life-extension and de-orbiting. Once robotic arm and fuel transfer technologies become more mature, more advanced services such as component upgrade and refueling will become viable options to increase the durability and competitiveness of GEO infrastructures, requiring satellite manufacturers to design and build upgradable and refuel-able spacecraft. Thus, as OOS technologies mature and service offering broadens, the traditional paradigm of long-lived bur poorly adaptable GEO satellites will shift towards a paradigm of maintainable and upgradable assets.

OOS is crucial for making the GEO environment more durable. The fundamental piece of any OOS infrastructure is a fleet of one or more servicers. A servicer is an autonomous or semi-autonomous spacecraft used to physically access the client satellites and provide them with services. However, servicers cannot solve alone the problem of sustainability in Earth's orbits: once a servicer's propellant tank gets empty, it becomes useless, must be discarded into a graveyard orbit, and thus becomes another piece of orbital debris. Thus, the fleet of GEO satellites will truly become sustainable once the OOS infrastructures serving them become sustainable.

Sustainable OOS infrastructures must involve other elements that are crucial to ensuring the long-term operations of the servicers, namely, orbital depots and resupply vehicles. These elements are key to establishing the supply chain of the commodities (e.g., propellant, spares) consumed by the servicers and the GEO satellites. The depots will store these commodities at fixed orbital locations and will be resupplied from Earth or other orbital depots by resupply vehicles. With the advent of sustainable OOS infrastructures, tools are needed to efficiently plan OOS missions.

OOS missions involving a single servicer can be solved as Traveling Salesman Problems: given some delta-V budget, the servicer must visit a sequence of client satellites with pre-known needs to maximize a particular metric, e.g., revenues from the visited satellites.

Such a problem can be made more complex by considering several servicers with different functions. For example, a servicer might be specialized in refueling satellites while another one might be specialized in upgrading them.

The Traveling Salesman Problem formulation and its extension, as described above, come limited when considering two additional characteristics of OOS:

1. As mentioned above, truly sustainable OOS will involve servicers that can be re-supplied in some commodities (e.g., propellant, spares) at depots so they can be operated over the long-term. With depots, servicers are not discarded at the end of a sequence of service tasks. They must keep enough propellant to go back to the depot and get refueled before providing the next sequence of service tasks. The interactions between depots and servicers and the in-space supply chain of commodities are not accounted for in the traditional Traveling Salesman Problem formulation. A more general formulation is required to accurately model complex OOS infrastructures involving various interacting elements.

2. Not all service needs are known ahead of time. Satellites may experience service needs on a random basis, e.g., the replacement of a failed component. To increase their competitiveness, OOS infrastructures will have to leverage the uncertain environment they will be operating in, by capturing additional revenues from unplanned service needs. In the Traveling Salesman Problem, the service needs are known before planning the missions of the servicers making it inadequate to plan the operations of OOS infrastructures in uncertain environments.

Chapter 2 is a literature review about past OOS analyses and space logistics methods that are relevant to the work presented in this thesis.

Chapter 3 presents the mission scenario, the model of the space logistic network based on which this work is developed, as well as the models of the spacecraft trajectories used to compute the costs of the transportation arcs of the network.

Chapter 4 develops a framework that models and optimizes the operations of sustainable OOS infrastructures – comprised of high-thrust servicers, depots, and resupply vehicles – dedicated to the servicing of a fleet of co-planar GEO satellites. The framework generalizes the Time-Expanded Generalized Multi-Commodity Network Flow (TE-GMCNF) problem formulation used in state-of-the-art space logistics studies to accurately model the OOS supply chain. The TE-GMCNF model is solved as a Mixed-Integer Linear Program (MILP). The RH decision making approach is then adapted to the TE-GMCNF model to account for uncertain service needs that arise over time among the GEO satellites. Two case studies successfully demonstrate the effectiveness of the framework for (1) short-term operational scheduling and (2) long-term strategic decision making for OOS architectures under diverse market conditions.

Chapter 5 generalizes the TE-GMCNF model proposed in chapter 4 so the operations of not only high-thrust servicers but also low-thrust and/or multimodal servicers can be optimized. The framework proposed in chapter 4 is also generalized by integrating user-defined trajectory models and optimizing the logistics operations with the propulsion technology and trajectory tradeoff in consideration. Several analyses are carried out to demonstrate the value of the proposed framework in automatically trading off the high- and low-thrust propulsion systems for both short-term operational scheduling and long-term strategic planning of OOS infrastructures.

Finally, chapter 6 generalizes the framework proposed in chapter 4 to the case of client satellites distributed across different orbits of various shapes and orientations. This is done by accounting for the relative dynamics of the nodes of the OOS network to compute the costs of the network arcs over simulation time. The time-varying orbital elements are inputted into a high-thrust trajectory optimization routine interfaced with the framework to accurately compute the cost of transportation of the servicers. Two case studies demonstrate the application of the generalized framework to the short-term operational scheduling and long-term strategic planning of on-orbit servicing infrastructures in GEO orbits.

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

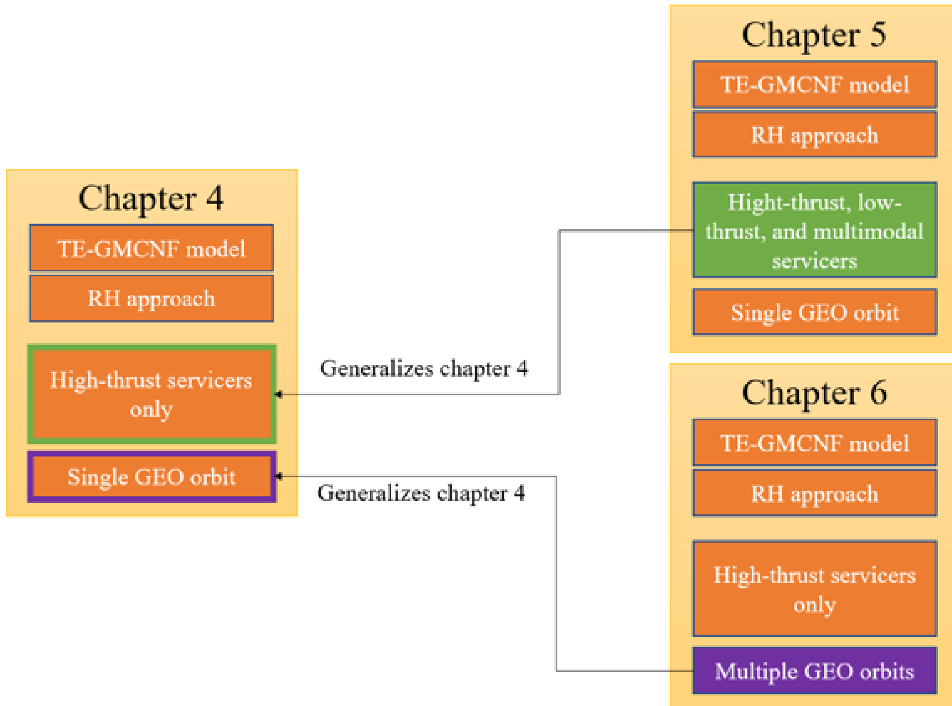

Figure 1.1: Dissertation overview and assumptions in chapters 4,5,6.

CHAPTER 2
MOTIVATION AND LITERATURE REVIEW

2.1 Motivation

2.1.1 Challenges faced by the GEO infrastructure

From telecommunications to weather monitoring and defense applications, GEO satellites have played a critical role as a key infrastructure of modern civilization, but they are also known for the difficulty of their maintenance due to their high altitude. Recently, a decrease in the replacement of the GEO satellites has been observed [1]. This indicates that, while GEO satellite operators are weighing the pros and cons between investing in large-scale LEO constellations and investing in high-throughput GEO satellite technologies [2], the fleet of operational GEO satellites is aging.

In addition, replacing a GEO satellite costs on the order of $150 million to $500 million [3] – to manufacture, insure, and launch these large spacecraft to the remote GEO. Until very recently, the traditional paradigm to maintain and upgrade this infrastructure consisted of replacing the outdated satellites with new assets designed to last 15 to 20 years. Besides the obvious issue that satellites designed to operate for that long cannot benefit from rapidly improving technologies, more profound issues exist related to the aversion to risk and innovation. Given the colossal amount of capital needed to deploy and operate space assets, this mindset is understandable but unsustainable if the GEO infrastructure is to become a major player in the cislunar space economy.

2.1.2 On-Orbit Servicing to address the challenges faced by the GEO infrastructure

On-Orbit Servicing (OOS) may be the tool needed to shift this mindset. At the time of this writing, OOS has been around for three decades starting with the maintenance of the

Hubble Space Telescope by the crews of the Space Shuttle. With the progress of space technologies, visionary U.S. agencies such as NASA and DARPA quickly took the lead with ambitious robotic servicing missions (e.g., NASA's OSAM 1 [4]; DARPA's Robotic Servicing of Geosynchronous Satellites [5]). Nowadays, as the cost of access to space dramatically decreases, incumbents and new entrants alike are developing innovative technologies and business models that will provide new game-changing services to the GEO infrastructure (e.g., refueling, upgrading, repositioning, etc.). With servicing spacecraft (aka servicers) regularly visiting GEO assets, satellite manufacturers and operators may change the way they traditionally do business, leading to enhanced resilience to competition and market fluctuations.

Although OOS has for a long time been the playground of governmental organizations, the private sector is now stepping in, developing servicers and robotic technologies to address the GEO servicing market. Northrop Grumman is leading the charge with their Mission Extension Vehicles (MEV), two of which are already providing life extension services to two of Intelsat's GEO satellites [6]. This is an exciting time for OOS proponents as technology and demand are aligning towards the success of this promising market, valued at $4.5 billion in cumulative revenues by 2028 [7].

In parallel to servicer development, the public and private sectors alike are actively focusing on the development of orbital depots and serviceable satellite designs to facilitate servicing. Orbital depots will store commodities, e.g., fuel, that can support the long-term operations of servicers and customer satellites. OrbitFab is actively working towards developing and deploying the first propellant depots in Earth orbits [8]. In addition, Lockheed Martins is actively working towards a serviceable design for the bus of the next generation of GPS satellites, located in MEO [9].

Developing a sustainable OOS infrastructure capable of capturing shares of the GEO servicing market will not be without risk, though. Large initial investments, uncertain demand, and yet-to-be-proven servicing technologies compound that risk. Meticulous strategic planning will be necessary to select the right technologies and business models that ensure a successful OOS endeavor. A typical tradeoff that decision-makers will likely face is between using high-thrust or low-thrust technologies to propel their servicers. Currently, the industry is leaning towards low-thrust propulsion. Northrop Grumman's MEVs for instance integrate Xenon Hall Thrusters [10]. Despite the high price of Xenon gas ($850/kg-$5000/kg in the past 15 years [11] [12]), this is a reasonable design given the low service demand and the efficiency and reliability of these thrusters. However, as orbital propellant depots [13] and their routine resupply by cheaper launch vehicles become a reality, and the increase in demand requires more responsive servicing operations, one will naturally question this decision and consider high-thrust propulsion, supported by regular servicer refueling, as a potential alternative. For example, Monomethyl Hydrazine (MMH) is an easily storable liquid propellant whose relatively low price ($170/kg in 2006 [14]) could make high-thrust servicers a competitive option.

To make valuable long-term design decisions, this emerging industry needs tools to explore the OOS design tradespace at the system level and guide technology portfolio management and roadmapping. Such tools will ensure the long-term resilience and robustness of the OOS businesses and their operations despite the uncertainties related to demand and competition.

2.2 Literature Review

2.2.1 Past OOS analyses

Despite the favorable turn that OOS is taking, this industry is still very much at its infancy. The tradespace of OOS is very large thus its exploration has been a challenging task. The existing GEO servicing literature can first be classified based on the type of OOS mission concept investigated:

1. One-to-one architectures in which one servicer (the servicing spacecraft) is serving

only one client satellite [15]. The first OOS missions will likely involve one-to-one architectures with servicers designed to provide one type of service such as station keeping. Northrop Grumman's on-going MEV-1 mission [6] is a one-to-one example. Infrastructure elements such as orbital depots may also be considered to further support the OOS operations;

2. One-to-many architectures in which one servicer provides services to a fleet of several client satellites [16] [17] [18] [19] [20] [21]. An orbital spare depot or refueling station may be considered to further support the OOS operations;

3. Many-to-many architectures in which several servicers provide services to a fleet of client satellites [22] [23] [24] [25]. Multiple orbital spare depots and refueling stations may be considered to further support the OOS operations.

For each OOS mission concept, the literature may then further be classified in two categories:

1. The analysis and design of accurate orbital trajectories associated with multi-transfer servicing operations, using either high-thrust [16] or low-thrust servicers [22]. These analyses usually consider high-fidelity models of perturbation forces to design the transfer trajectories;

2. The analyses of the operational strategy of a particular OOS infrastructure from a scheduling [17] and/or design standpoint [15] [18] [19]. Simulation [18] [20] [23], optimization [17] [19] [21] or a mixture of both [15] may be used to explore the value of the investigated OOS architecture by varying its operational scheduling and/or design.

   Despite a prolific OOS literature, the operational scheduling and system design of sustainable many-to-many OOS infrastructures has been largely unexplored. This is because this problem is complex and requires a sophisticated optimization problem formulation to handle that.

Leveraging the recent development in space logistics, this thesis aims to fill that gap by developing a new framework to optimize the OOS operations and system design under service demand uncertainties. The developed method will extend the state-of-the-art Mixed-Integer Linear Programming (MILP)-based space logistics modeling to address the unique challenges in OOS applications and integrate it with the dynamic RH decision making approach. The resulting framework enables both the short-term scheduling of OOS operations and the long-term decision making for the OOS systems design.

2.2.2 State-of-the-art space logistics methods

Space logistics has largely been explored in the literature to optimize the design and planning of complex human space exploration missions through heuristics methods [26], simulations [27], the graph theory [28], or network flow models [29] [30] [31] [32] [33] [34]. The method described in this thesis extends the latter to accurately model OOS operations. In the traditional space logistics network flow models, the formulation only specifies as inputs the demand of commodities at locations of interest such as Mars or the Moon. The optimizer then decides when and where to assign space vehicles in order to satisfy the demand. In the context of OOS, the servicers must stay a certain amount of time at the satellites' locations to provide the services. The traditional space logistics formulation does not capture this constraint and therefore a unique extension is needed for the OOS applications. The advantage of network flow models is that they can be modeled as MILP problems which can theoretically reach global optima.

In addition, the proposed framework must be able to perform the tradeoff between different types of propulsive options (e.g., high-thrust vs low-thrust trajectories) in an automated fashion. Previous works give important methodological pieces to take on the challenge of automated propulsion system tradeoff for the specific problem of OOS planning. This thesis uniquely combines them to generalize the state-of-the-art OOS logistics framework. First, Ref. [35] explores the automated tradeoff between high-thrust and low-thrust vehicles for the design of human space exploration campaigns. This was the first time that low-thrust technologies were modeled within space logistics analyses that leverage MILP as the globally optimal solution method. In Ref. [35], the piece-wise linear approximation of the low-thrust model is used along with a new variant of dynamic networks called event-driven networks. In the problem at hand, the event-driven networks are not needed because, unlike the cislunar transportation problem addressed in Ref. [35], the trajectories considered in this thesis are for Earth-orbiting satellites to transfer between similar orbits within the GEO orbital regime, which can be modeled as a reasonable number of predefined options using time-expanded networks. However, the piecewise linear approximation of non-linear low-thrust propellant consumption models is needed to enable their integration into a MILP. This idea of leveraging piecewise linear approximation of non-linear models in MILP was first proposed by Vielma et al. [36]; it was first introduced to space logistics formulation by Chen and Ho for the integration of non-linear infrastructure design into MILP [32], before Jagannatha and Ho used it to allow the optimizer to automatically perform the tradeoff between high- and low-thrust spacecraft in multi-mission space campaigns [35].

Another methodological piece leveraged in this chapter to properly embed the high-thrust/low-thrust tradeoff is the concept of multiarcs between the nodes of a network. This was first proposed by Ishimatsu et al. [29] and subsequently used in state-of-the-art space logistics methods [30] [31] [32] [33] [34] to model several space transportation options between the nodes of the space network. In this thesis, multiarcs are used to not only represent the transportation of commodities within different types of space vehicles but also to model the different servicer trajectory options available to the optimizer. For example, a servicer may have the option to fly on a 10-day or a 20-day rendezvous maneuver depending on the time it has left to reach and serve a customer satellite. The optimizer's choice of the transportation arcs in the final solution is ultimately driven by the tradeoff between servicing responsiveness and operating costs.

2.2.3 The Rolling Horizon decision making procedure

Finally, RH decision making, which is leveraged to design a long-term strategy for OOS, has been typically used to make decisions in a dynamic stochastic environment, usually characterized by uncertainties in demand and the resulting cost in forecasting this demand. More specifically, this technique consists in making a series of decisions repeatedly based on short- to mid-term forecast of future demand [37] [38]. An advantage of RH is to decompose a complex dynamic scheduling problem into smaller sub-problems whose combined optimal solutions yield a satisfying solution to the original problem at a lesser computational cost. In the literature, RH has been used for the optimal scheduling of Earth observing satellites [39] but never for the operational scheduling of OOS infrastructures.

CHAPTER 3
ON-ORBIT SERVICING NETWORK MODELING

This chapter introduces the mission scenario and assumptions related to the customer satellites and servicing infrastructure used throughout the thesis.

3.1 Mission Scenario

Although the developed framework can be applied to various OOS concepts, the following OOS concept is considered in this thesis as an example. A servicing company deploys high-thrust, low-thrust, and/or multimodal servicers, as well as orbital depots that store the commodities needed to support the operations of the servicers (e.g., propellant, spares). The servicers are equipped with robotic tools designed to perform the operations required for a specific set of services. Launch vehicles may resupply the depots and/or servicers with new commodities if needed. Whenever a GEO satellite requires a service, the OOS operator first decides whether to provide it, and, if so, dispatches the adequate servicer. After performing its task, the servicer then either goes back to its orbital staging location for storage, or flies to another customer satellite if need be. The notional services modeled in this chapter are inspection, refueling, station keeping, satellite repositioning, repair, mechanism deployment, and retirement. They fall in either of two categories: random (e.g., unplanned such as component failure), or deterministic (e.g., pre-planned such as refueling).

3.2 Customer and Servicing Infrastructure Assumptions

This section presents the general modeling of the fleet of customer satellites and their service demands, the modeling of the OOS architecture, the modeling of the phasing maneuvers performed by the servicers, and an overview of the OOS logistics static network.

3.2.1 Customer satellites and orbital service needs

The customer satellites are assumed to be distributed along similar circular orbits in the GEO orbital regime. Chapter 4 and chapter 5 restrict these orbits to a single circular, coplanar orbits. This assumption simplifies the MILP modeling of the OOS operations since considering different orbits would require accounting for the phasing angles between the satellites at any time during the optimization, thus adding unnecessary complexity for some applications of interest such as GEO servicing. Chapter 6 relaxes the assumption of a single circular, co-planar orbit to model more realistic servicing scenarios.

Other parameters of the customer satellites include their masses, the type of propellant they use for station keeping, and what type of life extension service they require (i.e., refueling or station keeping) if such a need arises during their lifetimes. In addition, each defined satellite is specified parameters that model the characteristics of their potential service needs.

Seven types of service needs are considered in this dissertation. Three of them are assumed deterministic, that is, they occur on a regular basis whereas the last four are assumed to arise randomly. Table 3.1 and Table 3.2 define the deterministic and random service needs modeled in this work. The main parameters used to model the deterministic and random service needs are defined in Table 3.3.

Table 3.1: Definition of the deterministic service needs (i.e., planned ahead of time).

	Inspection	**Refueling**	**Station Keeping**
Definition	The servicer performs a proximity maneuver near the satellite, and inspects its condition without docking to it.	The servicer rendezvouses and docks to the satellite to top up its tank with additional propellant.	The servicer rendezvouses and docks to the satellite to perform station-keeping maneuvers in place of the satellite.

Table 3.2: Definition of the random service needs (i.e., not planned ahead of time).

	Repositioning	**Retirement**	**Repair**	**Mechanism Deployment**
Definition	The servicer rendezvouses and docks to the satellite to change its orbital slot.	The servicer rendezvouses and docks to the defunct satellite to transport it to some graveyard orbit.	The servicer rendezvouses and docks to the satellite, and replaces the defective parts with spare ones.	The servicer rendezvouses and docks to the satellite, and unlocks its stuck appendages.

3.2.2 On-orbit servicing architectural elements

The OOS infrastructures modeled in this chapter include the servicers and their tools, orbital depots, and launch vehicles. A servicer is defined here as a fully- or semi-autonomous robotic spacecraft that uses tools to provide services to customer satellites. An OOS infrastructure needs at least one such servicer to physically access the customer satellites and provide them services. The model of a servicer is captured by four sets of parameters defined in Table 3.4. Chapter 4 and chapter 6 assume that the servicers use high-thrust propulsion technologies. Thus, the rocket equation can be integrated in a MILP formulation as a linear equation relating the total servicer mass and its propellant consumption. Chapter 5 relaxes this assumption by modeling low-thrust servicers and their trajectories.

The servicers must also have the right tool(s) to provide the services. A mapping between the service needs presented in subsection 3.2.1 and four notional tools is presented in Table 3.5. Note that the optimization framework is flexible enough to consider any kind of user-defined tools. In this thesis, a service cannot be provided by several types of tools, but some tools can be used to provide several types of services. The parameters used to model a tool are its mass and its cost (development, manufacturing, etc.).

In addition to the servicers and their tools, an OOS infrastructure may use one or more depots that act as in-space propellant depots and warehouses, and robotic manutention systems (e.g., the International Space Station). Their functions are to store commodities needed to provide services such as refueling and repair, and to transfer commodities between any two vehicles (e.g., between the payload bay of a launch vehicle and a servicer). A depot offers the advantage of more responsive services but at the cost of a more expensive OOS infrastructure. The main parameters of a depot are its dry mass (to calculate how much it costs to deploy it in orbit), the capacities of its payload bays (e.g., how much propellant and spares it can store), its manufacturing and operating costs, and its own propellant consumption in order to perform station keeping. Two important assumptions related to depots are made in this thesis: (1) depots never execute orbital maneuvers, and (2) they are deployed at some orbital slots (referred to as OOS parking nodes). Chapters 4 and 5 assume that the orbital depots are deployed along the same orbital ring as the fleet of customer satellites. This latter assumption facilitates the modeling of the OOS scheduling problem as a MILP. Chapter 6 relaxes this assumption and allows the orbital depots to be deployed along their own orbits.

A launch vehicle is modeled with three main parameters: its launch frequency (assumed deterministic in this chapter); its launch price tag (in $/kg); and its payload capacity to launch commodities to Geosynchronous Transfer Orbit (GTO) or GEO. In this thesis, launch vehicles are assumed to rendezvous with depots and/or servicers at the OOS parking nodes.

Finally, the consumables considered in this thesis are the spare parts, bipropellant, monopropellant and electric propulsion engine propellant. The spare parts are assumed to be an undifferentiated commodity, that is, a spare transponder is the same thing as a spare antenna. Several types of propellant were defined to allow more flexibility in the modeling of the servicers and customer satellites. For example, nowadays, GEO satellites use either monopropellant thrusters or ion engines for station keeping, while orbital maneuvers such as apogee kicks (to transfer from GTO to GEO) or inclination change are done primarily using monopropellant or bipropellant.

Table 3.3: Definition of the random service needs (i.e., not planned ahead of time).

Parameter	**Applies to deterministic and/or random needs?**	**Definition**
Revenue	Deterministic and random	Revenue received by the OOS infrastructure for providing a service in response to some service need.
Delay penalty cost	Deterministic and random	How much it costs daily to the OOS operator to delay the beginning date of the service.
Service duration	Deterministic and random	The time it takes to provide a service.
Service window	Deterministic and random	Time interval within which the service must start, provided that the OOS operator decides to provide the service.
Interoccurence time	Deterministic only	Used to generate service needs regularly spaced in time.
Mean interoccurence time	Random only	Used to model service need arrivals as a Poisson process.

Table 3.4: Parameters to model the servicers.

Parameter set	**Definition**	**Examples of parameters**
Tool parameters	Specify how many tools the servicer is equipped with and of which type those tools are.	Tools integrated to the servicer's bus (cf Table 3.5)
Orbital transfer parameters	Specify the performance of the servicer in executing phasing maneuvers.	Specific impulse (Isp); propellant tank capacity; dry mass; thrust level
Cost parameters	Specify how much it costs the OOS company to develop, manufacture, and operate a servicer	Development costs; operating costs
Payload capacity parameters	Capacities of the servicer's payload bays to provide services that require commodities (e.g., refueling, repair)	Capacity of the servicer's tank used to transport customer satellites' propellant; payload capacity to transport spare parts

Table 3.5: Notional service-tool mapping

	T1: Refueling apparatus	**T2: Observation sensors**	**T3: Dexterous robotic arm**	**T4: Capture mechanism**
Inspection		x		
Refueling	x			
Station Keeping				x
Repositioning				x
Retirement				x
Repair			x	
Mechanism Deployment			x	

CHAPTER 4

FRAMEWORK FOR MODELING AND OPTIMIZATION OF ON-ORBIT SERVICING OPERATIONS UNDER DEMAND UNCERTAINTIES

4.1 Modeling

4.1.1 Scope

This chapter assumes the following:

- The customer satellites and orbital depots are co-located along a single, circular orbit;
- The servicers are equipped with high-thrust chemical engines;
- The servicers use high-thrust phasing maneuvers to move from one node of the network to another.

Figure 4.1 gives an overview of the OOS logistics network used in this chapter.

4.1.2 High-Thrust Phasing Maneuver Model

In this subsection is presented a trajectory model used in the on-orbit servicing framework to compute the arcs of the network. Phasing maneuver-based rendezvous are two-impulse trajectories between two different positions on the same circular orbit. The semi-major axis of the phasing orbit is calculated by minimizing the delta V under altitude and time-of-flight constraints. Figure 4.2 illustrates the situation when the semi-major axis of the servicer phasing trajectory is larger than the GEO orbit radius. However, the algorithm proposed here can also find optimal semi-major axes smaller than the GEO orbit radius.

The radius of the GEO orbit is denoted by r. Similarly, r_{crit} is the radius of the "forbidden flight zone" defined around the Earth where the servicer is not allowed to fly. The

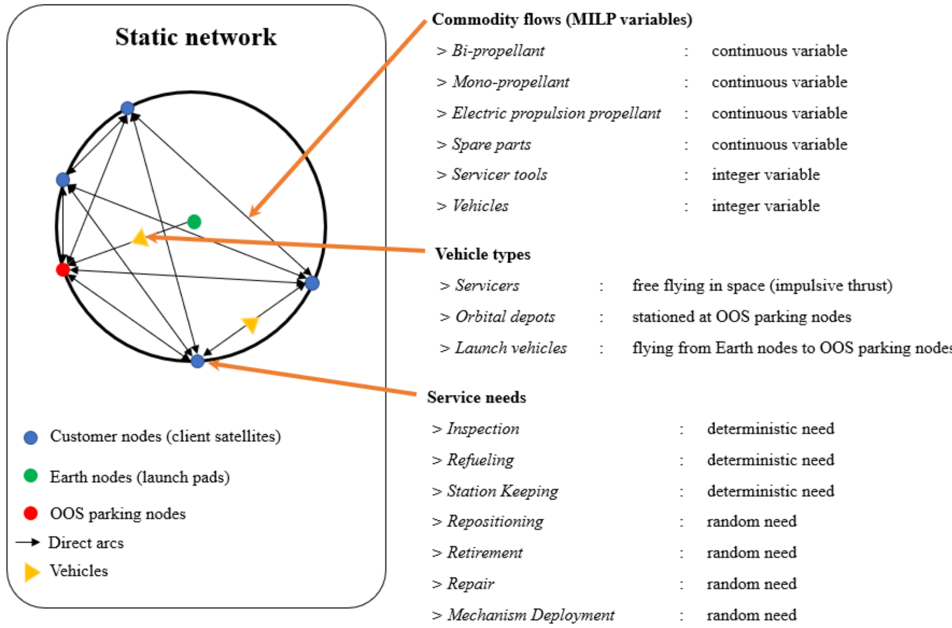

Figure 4.1: High-level overview of the OOS logistics network used in developing the OOS optimization framework.

initial relative angle from the servicer to the GEO satellite is $\Delta\theta_0 \in [0, 2\pi)$. Similarly, the initial relative angle from the GEO satellite to the servicer is $\alpha = 2\pi - \Delta\theta_0$ and is used instead of $\Delta\theta_0$ in the equations presented below.

In order to define the travel time of the servicer and satellite on their respective orbit, two integers are defined, $k_1 \geq 1$ and $k_2 \geq 0$, corresponding to the number of complete orbits they fly before rendezvousing. From the point of view of the servicer, the travel time is

$$t_{travel} = k_1 T_1 = 2\pi \; k_1 \sqrt{\frac{a^3}{\mu}} \tag{4.1}$$

where T_1 and a are the period and semi-major axis of the phasing orbit respectively, and μ is the gravitational parameter of the Earth. From the point of view of the satellite, the travel time is

$$t_{travel} = \frac{\alpha}{2\pi} T + k_2 T = \left[\frac{\alpha + 2\pi k_2}{2\pi}\right] T = (\alpha + 2\pi k_2)\sqrt{\frac{r^3}{\mu}} \tag{4.2}$$

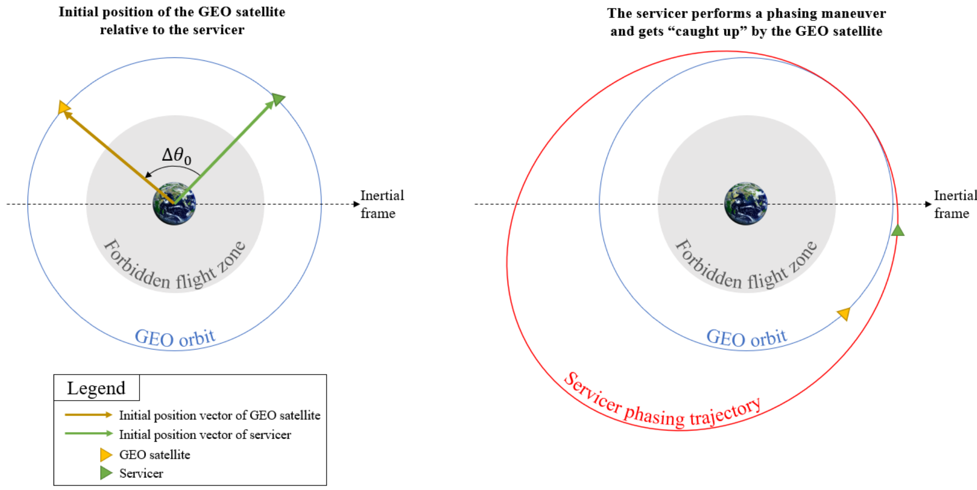

Figure 4.2: Example of rendezvous between a servicer and a GEO satellites by making use of a phasing maneuver.

where T is the period of the orbit of the satellite. Equaling the above two equations gives an expression for a as a function of k_1 and k_2:

$$a = \left[\frac{\alpha + 2\pi k_2}{2\pi k_1} \right]^{2/3} r \tag{4.3}$$

The altitude constraint can then be written as

$$a \geq \frac{r + r_{crit}}{2} \tag{4.4}$$

Solving for k_1 and k_2 is a two-step process. One can see from Equation 4.2 that the travel time can be expressed as a function of only one unknown, k_2, from the perspective of the satellite. Thus, this equation is used to find the k_2 values that satisfy the time-of-flight constraint. For each tested value of k_2, values of k_1 are tested using Equation 4.3 and v. If the pair

$$(k_1, k_2)$$

satisfies both the time-of-flight and altitude constraints, it is kept as candidate solution. Once all such feasible pairs have been found, the one with the lowest delta V is kept as solution. The delta V is calculated using:

$$\Delta V = 2\left|\sqrt{\frac{\mu}{r}} - \sqrt{\mu\left(\frac{2}{r} - \frac{1}{a}\right)}\right| \tag{4.5}$$

4.2 Methods

This section introduces the three important methodological pieces making up the proposed OOS optimization framework. It is desired to (1) accurately model the OOS operations over the time dimension, (2) inform long-term decision making while considering uncertainties inherent to the random service needs (e.g., repair), and (3) model the OOS scheduling problem as a MILP of which a global optimum can theoretically be reached, given enough time to solve it. Subsection 4.2.1 describes how the static network presented in chapter 3 is expanded over time. Next, subsection 4.2.2 presents the traditional space logistics MILP formulation and its extension to accurately model OOS operations. Finally, subsection 4.2.3 describes how the traditional RH decision making methodology is adapted in the context of OOS to allow for sustainable architectural decisions under uncertainties.

4.2.1 Static network time expansion

This subsection shows how one can model the OOS operations over the time dimension by replicating the static OOS logistics network at predefined time steps. This may be done with a very fine time discretization at a cost of a large optimization time. Indeed, the number of MILP variables increases linearly with the number of time steps at which the static network is replicated. On the other hand, a very coarse time discretization may lead to inaccuracies in the OOS operations. Based on the application and the time available to run the analyses, a coarse or fine time discretization may be selected. This is done through the definition of three parameters: (1) the length Δt of so-called spaceflight time steps that are used to accurately model the transportation of vehicles over the time-expanded network (also referred to as dynamic network in this chapter); (2) a service time interval T; and (3) the number n of spaceflight time steps within the time interval T. The following introduces the concept of each parameter.

The parameter Δt controls the coarseness of the dynamic expansion of the static network. It is the least common multiple of all time steps defined in the dynamic network.

The parameter T is chosen at most as large as the greatest common divisor of the durations of the considered services. For example, if refueling services are provided in 20 days and inspection services in 30 days, then the greatest common divisor would be 10 days and T would need to be chosen less than or equal to 10 days. Note that the typical service durations found in the literature are 10-30 days [17] [21] which allow for a naturally coarse time discretization. The static network is then replicated every time interval T.

The parameter n defines the number of spaceflight time steps per interval T that are used to accurately represent the finite flights of the servicers. They represent the first Δt-long time steps defined within each time interval T. The remaining time step of length $T - n\Delta t$, called service time step, is used exclusively for the provision of services, which means that vehicles cannot fly during that time step. Note that the high-thrust trajectories are designed with a time of flight not exceeding ΔT justifying the considered time-expanded topology of the network.

Figure 4.3 depicts a notional time-expansion of a simple 3-node static network for $\Delta t =$ 1 day, $T = 10$ days, and $n = 2$. This means that the servicers can fly over the network only between $t = 0$ and $t = 2$, between $t = 10$ and $t = 12$, etc. The time steps defined between $t = 2$ and $t = 10$, between $t = 12$ and $t = 20$, etc., are only used for the provision of services. This is illustrated by the bold, yellow direct arc on Figure 4.3: between $t = 0$ and $t = 2$, a servicer is launched from Earth to the OOS parking node via a launch vehicle, and then flies on its own from the OOS parking node to the customer node; between $t = 2$ and $t = 20$, the servicer provides a service before going back to the OOS parking node for storage at $t = 20$. Note that the service needs may only arise at $t = 0, 10, 20$, etc.

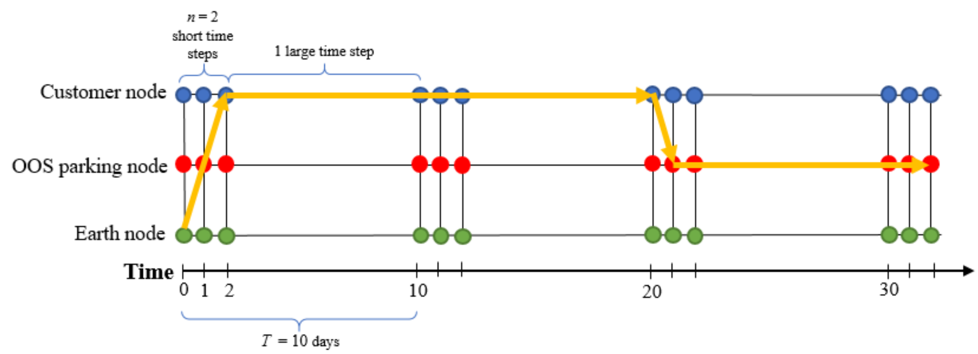

Figure 4.3: Notional time-expanded network and servicer path with $\Delta t = 1$ day, T = 10 days, and n = 2.

4.2.2 On-Orbit Servicing Logistics Mathematical Formulation

This subsection introduces the next methodological piece making up the proposed OOS optimization framework: the mathematical OOS logistics optimization formulation. In particular, the MILP model used for traditional space logistics problems (e.g., interplanetary human space mission design) is presented, the reasons for which that method cannot be used for OOS operations are given, and the proposed extension for OOS application is presented.

Traditional Space Logistics Mixed Integer Linear Programming Formulation

The traditional space logistics problem is formulated as a MILP model that builds upon the dynamic network formalism presented in Figure 4.3. Let's consider a time-expanded network graph $\mathcal{G}$ made up of a set of nodes $\mathcal{N}$ and a set of direct arcs $\mathcal{A}$. The latter include both transportation arcs that connect different nodes and holdover arcs that connect a node with itself. Each arc has an index (v, i, j), meaning that vehicle v flies from node i to node j. Additionally, the set of vehicles and time steps are denoted $\mathcal{V}$ and $\mathcal{T}$ respectively.

In the mathematical formulation presented here, the distinction is made between the vehicle commodity and the other commodities (cf Figure 4.1). The vehicle commodity flowing along arc (v, i, j) at time step t is denoted by the nonnegative scalar $y^{\pm}_{vijt}$. The definition is similar for the other commodities denoted by the $p \times 1$ column vector $\mathbf{x}^{\pm}_{vijt}$ of nonnegative scalars, where p is the number of non-vehicle commodities. The commodities flow over the arcs and are split between node outflow $y^{+}_{vijt}/\mathbf{x}^{+}_{vijt}$ (i.e., they leave node i) and node inflow $y^{-}_{vijt}/\mathbf{x}^{-}_{vijt}$ (i.e., they enter node j). The vehicle commodity flow variables $y^{\pm}_{vijt}$ are integer. The non-vehicle commodity flow variables are either integer or continuous: the set $\mathcal{K}_I$ denotes the commodities that are integer, whereas $\mathcal{K}_C$ denotes the commodities that are continuous. Figure 4.1 shown earlier gives the natures of the variables that represent the commodities considered in this chapter.

The cost coefficients are denoted by the scalar ${c'}^{+}_{vijt}$ for the vehicle commodity, and by the $1 \times p$ row vector $\mathbf{c}^{+}_{vijt}$ for the other commodities. Each node i of the static network is assigned at each time step a scalar vehicle demand ${d'}_{ivt}$ and a $p \times 1$ demand vector $\mathbf{d}_{it}$ for the other commodities, where demand and supply are assigned non-positive and non-negative values respectively. The MILP formulation presented below is based upon Ref. [32].

Minimize

$$J = \sum_{t \in \mathcal{T}} \sum_{(v,i,j) \in \mathcal{A}} \left\{ \mathbf{c}^{+}_{vijt} \mathbf{x}^{+}_{vijt} + {c'}^{+}_{vijt} y^{+}_{vijt} \right\} \tag{4.6}$$

$$\sum_{(v,j):(v,i,j) \in \mathcal{A}} \mathbf{x}^{+}_{vijt} - \sum_{(v,j):(v,j,i) \in \mathcal{A}} \mathbf{x}^{-}_{vji(t-\Delta t)} \leq \mathbf{d}_{it} \quad : \quad t \in \mathcal{T},\ i \in \mathcal{N},\ v \in \mathcal{V} \tag{4.7}$$

$$\sum_{(v,j):(v,i,j) \in \mathcal{A}} y^{+}_{vijt} - \sum_{(v,j):(v,j,i) \in \mathcal{A}} y^{-}_{vji(t-\Delta t)} \leq d'_{ivt} \quad : \quad t \in \mathcal{T},\ i \in \mathcal{N},\ v \in \mathcal{V} \tag{4.8}$$

$$Q_{vij} \begin{bmatrix} \boldsymbol{x}^{+}_{vijt} \\ s_v y^{+}_{vijt} \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}^{-}_{vijt} \\ s_v y^{-}_{vijt} \end{bmatrix} \quad : \quad t \in \mathcal{T}, (v,i,j) \in \mathcal{A} \tag{4.9}$$

$$H_{vij} x^{+}_{vijt} \leq \boldsymbol{e}_v y^{+}_{vijt} \quad : \quad t \in \mathcal{T}, (v,i,j) \in \mathcal{A} \tag{4.10}$$

$$\begin{cases} x^{\pm}_{vijt} \geq 0_{p\times 1} & \text{if } t \in \mathcal{L}_{ij} \\ x^{\pm}_{vijt} = 0_{p\times 1} & \text{otherwise} \end{cases} \quad : t \in \mathcal{T}, (v,i,j) \in \mathcal{A} \tag{4.11}$$

$$\boldsymbol{x}^{\pm}_{vijt} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{bmatrix}^{\pm}_{vijt} \quad \text{where } x_1, x_2 \ldots x_p \in \mathbb{R}_{+} \text{or } \mathbb{Z}_{+} \quad : \quad t \in \mathcal{T}, (v,i,j) \in \mathcal{A} \tag{4.12}$$

$$y^{\pm}_{vijt} \in \mathbb{Z}_{+} \quad : \quad t \in \mathcal{T}, (v,i,j) \in \mathcal{A} \tag{4.13}$$

Table 4.1, Table 4.2, Table 4.3 define the indices, variables and parameters appearing in the MILP formulation. Table 4.4 explains the meaning of each equation appearing in the MILP formulation.

Table 4.1: Indices in the MILP formulation.

Name	**Definition**
v	Vehicle index
i, j	Node index
t	Time index

Table 4.2: Variables in the MILP formulation.

Name	Definition
$\mathbf{x}^{\pm}_{vijt}$	Commodity inflows/outflows. Commodities in $\mathbf{x}^{\pm}_{vijt}$ are integer or continuous variables based on the commodity type. For example, servicer tools are modeled as integers whereas propellant is modeled as continuous.
$y^{\pm}_{vijt}$	Number of vehicles of type v flying from node i to node j at time t. Integer variable.

Extension of the Traditional Space Logistics Formulation for On-Orbit Servicing

Despite the effectiveness of the traditional space logistics formulation in optimizing complex interplanetary space missions [29] [30] [31] [32] [33] [34], it cannot properly model OOS operations because it does not specify how long a vehicle must remain at a node of the network. Indeed, servicers must be providing services for some duration to the customer satellites. To that purpose, two new sets of binary variables and the concept of service window are introduced. OOS-specific constraints are then defined to properly model OOS operations. Finally, the objective function used in this chapter is presented.

The traditional space logistics problem is formulated as a MILP model that builds upon the dynamic network formalism presented in Figure 4.3. Let's consider a time-expanded network graph $\mathcal{G}$ made up of a set of nodes $\mathcal{N}$ and a set of direct arcs $\mathcal{A}$. The latter include both transportation arcs that connect different nodes and holdover arcs that connect a node with itself. Each arc has an index (v, i, j), meaning that vehicle v flies from node i to node j. Additionally, the set of vehicles and time steps are denoted $\mathcal{V}$ and $\mathcal{T}$ respectively.

In the mathematical formulation presented here, the distinction is made between the vehicle commodity and the other commodities (cf Figure 4.1). The vehicle commodity flowing along arc (v, i, j) at time step t is denoted by the nonnegative scalar $y^{\pm}_{vijt}$. The

Table 4.3: Parameters in the MILP formulation.

Name	Definition
$\mathbf{d}_{it}, \quad d'_{ivt}$	The demand of commodities and vehicles. In the OOS context, $\mathbf{d}_{it}$ represents the commodities needed to provide services (e.g., propellant for refueling services).
Δt_{ij}	The duration of the arc between node i and node j.
s_v	Structure mass of vehicle v.
$\mathbf{e}_v$	Vehicle design parameters, including payload and propellant capacities.
$\mathbf{c}^{+}_{vijt}$	Cost coefficients of the commodities other than the vehicle commodity.
c'^{+}_{vijt}	Cost coefficients of the vehicle commodity.
Q_{vij}	Commodity transformation matrix.
H_{vij}	Concurrency constraint matrix.
$\mathcal{L}_{ij}$	Index sets for the time steps at which launches from Earth occur.

definition is similar for the other commodities denoted by the $p \times 1$ column vector $\mathbf{x}^{\pm}_{vijt}$ of nonnegative scalars, where p is the number of non-vehicle commodities. The commodities flow over the arcs and are split between node outflow $y^{+}_{vijt}/\mathbf{x}^{+}_{vijt}$ (i.e., they leave node i) and node inflow $y^{-}_{vijt}/\mathbf{x}^{-}_{vijt}$ (i.e., they enter node j). The vehicle commodity flow variables $y^{\pm}_{vijt}$ are integer. The non-vehicle commodity flow variables are either integer or continuous: the set $\mathcal{K}_I$ denotes the commodities that are integer, whereas $\mathcal{K}_C$ denotes the commodities that are continuous. Figure 4.1 shown earlier gives the natures of the variables that represent the commodities considered in this chapter.

The cost coefficients are denoted by the scalar c'^{+}_{vijt} for the vehicle commodity, and by the $1 \times p$ row vector $\mathbf{c}^{+}_{vijt}$ for the other commodities. Each node i of the static network is

Table 4.4: Meaning of the equations in the MILP formulation.

Equation	Name	Description
Equation 4.6	Objective function	Minimize total mission cost
Equation 4.7	Mass balance constraint	The mass balance of the commodity flows (other than the vehicle commodity) into and out of node i.
Equation 4.8	Mass balance constraint	The mass balance of the vehicle commodity flow) into and out of node i.
Equation 4.9	Flow transformation constraint	Transformation of the payloads during the flight along the arcs (e.g., propellant burning).
Equation 4.10	Flow concurrency constraint	Concurrency constraints of the flow related to the vehicles, such as payload bay and propellant tank capacities.
Equation 4.11	Flow bound constraint	Time window (only when a time window opens is the commodity flow permitted).

assigned at each time step a scalar vehicle demand d'_{ivt} and a $p \times 1$ demand vector $\mathbf{d}_{it}$ for the other commodities, where demand and supply are assigned non-positive and non-negative values respectively. The MILP formulation presented below is based upon Ref. [32].

i. OOS-specific binary variables

The OOS-specific binary variables are introduced to allow the definition of additional constraints that properly assign services to servicers, and force servicers to remain at customer satellites' locations wherever services must be provided. The two new sets of binary variables modeling this are the *service assignment variables* and the *servicer dispatch variables*.

- Service assignment variables and service window

The service assignment variables, denoted $h_{sv\tau}$, are defined to specify at what date τ a given service s must start and which servicer v must provide it. More specifically, $h_{sv\tau} = 1$ if the optimizer requires servicer v to start service s at the date τ. Before proceeding further with the description, four new index sets associated with the indices v, s, τ must be introduced. The set of servicers $\mathcal{V}_{serv}$, associated with index v, is a subset of the set of vehicles $\mathcal{V}$. The set of services $\mathcal{S}$ includes all combined customer satellites' service needs occurring over the planning horizon (PH). The set of services $\mathcal{S}_i$ is a subset of $\mathcal{S}$ that corresponds to the services occurring at customer node i over the PH. The index associated with both $\mathcal{S}$ and $\mathcal{S}_i$ is s. The index set associated with τ is dependent upon the service index s and is denoted $\mathcal{W}_s$. It comprises every time step defined in the dynamic network at which service s may start. Thus, $\mathcal{W}_s$ is a subset of the set of time steps $\mathcal{T}$. It is related to a core concept of the proposed formulation, called service window, which is presented below.

Whenever a service need occurs at a given customer satellite, the OOS operator may be given some flexibility to select the date at which to dispatch a servicer to the satellite in need. This is allowed through the definition of a service window per each service need s that arises over the PH. For instance, let's assume that an OOS operator and a satellite operator agree on a 10-day service window whenever a refueling service need arises. Assume now that such a refueling service need, indexed by s_0, arises at that satellite at the date $t = 20$. The OOS operator decides to dispatch a servicer v_0 to provide the service s_0 at some date τ between $t = 20$ and $t = 30$. The optimizer embedded in the proposed framework would play the role of the OOS operator and decide at which date the service should start.

Figure 4.4 illustrates the above example with a simple dynamic network comprised only of the customer node where the service need s_0 arises. The Δt, T, and n parameters are 1 day, 10 days, and 2 respectively. Due to the discrete nature of the time-expanded network, there are only a few dates at which the optimizer can dispatch servicer v_0 to the satellite in need. In this work, it was decided to offset the service window by the n spaceflight time steps so that the servicer would have time to fly over the network between $t = 20$ and $t = 22$ and reach the satellite for the first service opportunity defined at $\tau = 22$ on Figure 4.4. In this work, the service opportunities (depicted by the red dots on Figure 4.4) are defined at the last of the spaceflight time steps encompassed within the service window. Each of these service opportunities is then given a service assignment variable whose value is decided by the optimizer. The time set $\mathcal{W}_{s_0}$ associated with service need s_0 would then be 22,32. This example illustrates faithfully what is happening under the hood of the proposed optimization framework, and can easily be generalized to more complex situations.

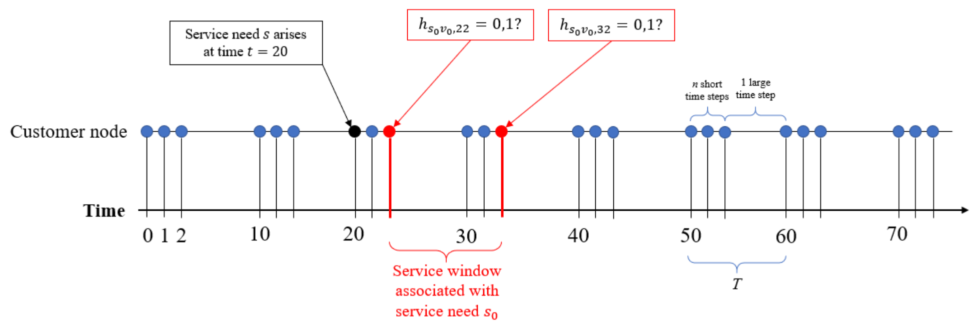

Figure 4.4: Illustration of the concept of service window and its relationship with the service assignment variables $h_{sv\tau}$ (Δt = 1 day, T = 10 days, and n = 2).

- Servicer dispatch variable

The service assignment variables previously introduced cannot be used to enforce the presence of a servicer at a given node since they are defined for subsets only (i.e., $\mathcal{W}_s$) of the entire set of time steps $\mathcal{T}$. To remedy this, the servicer dispatch variables, denoted b_{svt}, are defined for each time step defined in $\mathcal{T}$. Note that the time index of these variables is different from that of the service assignment variables (t instead of τ) to indicate that they are indexed over $\mathcal{T}$ and not over $\mathcal{W}_s$. At any time step t where $b_{svt} = 1$, servicer v must be in the process of providing service s. The optimizer sets it to 0 otherwise.

The servicer dispatch variables are related to the service assignment variables through the equality constraint:

$$b_{svt} = \sum_{\tau \in \mathcal{W}_s} h_{sv\tau}\beta_{s\tau t} \quad : \quad v \in \mathcal{V}_{serv}\ ,\ s \in \mathcal{S},\ t \in \mathcal{T} \tag{4.14}$$

The binary parameters $\beta_{s\tau t}$ defined in Equation 4.14 are automatically generated by the OOS optimization framework before each PH optimization based on the input data related to the services. Note that these parameters are indexed by both $t \in \mathcal{T}$ and $\tau \in \mathcal{W}_s$. Essentially, the binary parameter $\beta_{s\tau t}$ captures the duration of service s and at which time steps t the service has to be provided if the OOS operator decides to start the service at the date τ.

The above example used to describe the service assignment variables (cf Figure 4.4) is extended here to give more insight into Equation 4.14. From the situation illustrated in Figure 4.4, Equation 4.14 would be specialized into Equation 4.15. The notations from Figure 4.4 are kept for consistency. Also note that $\mathcal{W}_{s_0}$ was replaced with {22,32} in Figure 4.4.

$$b_{s_0 v_0 t} = \sum_{\tau \in \{22,32\}} h_{s_0 v_0 \tau} \beta_{s_0 \tau t} = h_{s_0 v_0,22} \beta_{s_0,22,t} + h_{s_0 v_0,32} \beta_{s_0,32,t} \quad : \quad t \in \mathcal{T} \tag{4.15}$$

Next, Table 4.5 illustrates what would be in this notional example the sequence of 1's and 0's for the $\beta_{s\tau t}$ parameters. In order to generate these parameters, it is assumed that the refueling service s_0 lasts for 40 days. Note that the 1's in Table 4.5 do not span 40 days; instead, this 40-day service is interpreted as over four service time steps (from after day 22 through right before day 60). This modeling is used to have the end of a service period match with the beginning of a service time interval T so that, when the servicer is done providing a service, it can benefit from the transportation arcs defined over the n spaceflight time steps.

Table 4.5: Notional sequence of binary parameters $\beta_{s\tau t}$ used to relate the $\mathbf{b_{svt}}$ variable with the $h_{sv\tau}$ variables.

Time	0	1	2	10	11	12	20	21	22	30	31	32	40	41	42	50	51	52	60	61	62	70	71	72
$\beta_{s_0,22,t}$	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
$\beta_{s_0,32,t}$	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0

Through the service assignment variables $h_{sv\tau}$, the optimizer then "turns on" the sequence of binary parameters that is optimal. The selected sequence of binary parameters is then assigned to the servicer dispatch variables b_{svt}. Figure 4.5 illustrates the selection of the sequence of binary parameters through $h_{sv\tau}$. Note that the optimizer is left with the choice not to provide a service which means that the $h_{sv\tau}$ variables in Equation 4.14 may all be set to 0.

Figure 4.5: Illustration of how the servicer dispatch variable β_{svt} are assigned a value by selecting a sequence of binary parameters $\beta_{s\tau t}$ through the service assignment variables h_{svt}.

ii. OOS-specific constraints

The OOS-specific variables introduced in the previous section are used to define three additional constraints that are appended to the traditional space logistics formulation. The first constraint given by Equation 4.16 specifies that each service s may be scheduled at most once.

$$\sum_{v \in \mathcal{V}_{serv}} \sum_{\tau \in \mathcal{W}_s} h_{sv\tau} \leq 1 \quad : \quad s \in \mathcal{S} \tag{4.16}$$

The second constraint given by Equation 4.17 specifies that at most one service may be provided to a customer satellite per time step. This means that if a customer satellite experiences a refueling service need and a repair service need at the same time, the OOS infrastructure will provide at most one of them. This constraint essentially prevents the clustering of servicers at a customer satellite in order to ensure the safest possible operations. The index set $\mathcal{N}_c$ in Equation 4.17 is a subset of the set of nodes $\mathcal{N}$ that includes only the customer nodes.

$$\sum_{v \in \mathcal{V}_{serv}} \left\{ \sum_{s \in \mathcal{S}_i} b_{svt} \right\} \leq 1 \quad : \quad i \in \mathcal{N}_c \, , \; t \in \mathcal{T} \tag{4.17}$$

The final constraint given by Equation 4.18 is critical for an accurate modeling of the OOS operations as it forces servicers at customer nodes for the entire durations of the services. This is done through the definition of a set of binary parameters γ_{sk} that map a tool k to service s as given in Table 3.5. For example, if service s is a refueling service, then $\gamma_{sk} = 1$ for k being the refueling apparatus tool whereas $\gamma_{sk} = 0$ otherwise. The index set of the tool commodities is denoted $\mathcal{K}_{tools}$ and is a subset of the set $\mathcal{K}_I$ of non-vehicle integer commodities. The servicers are forced at customer nodes by requiring the use of their tools for services.

$$x^{+}_{viitk} \geq \sum_{s \in \mathcal{S}_i} \gamma_{sk} b_{svt} \quad : \quad v \in \mathcal{V}_{serv} \, , \; i \in \mathcal{N}_c, \; t \in \mathcal{T}, \; k \in \mathcal{K}_{tools} \tag{4.18}$$

iii. OOS-specific objective function

Although most literature in space logistics has focused on the launch mass as the objective function, different objective functions may be defined based on the true motivations of the OOS venture. This chapter assumes that the modeled OOS company wishes to maximize the profits over a planning horizon. The profits are defined as the difference between the revenues generated by the provision of the services and the costs incurred by the infrastructure. The revenues are captured through the service assignment variables $h_{sv\tau}$ and revenue parameters r_s defined for each service. The revenues are modeled by Equation 4.19.

$$J_{revenues} = \sum_{v \in \mathcal{V}_{serv}} \sum_{i \in \mathcal{N}_c} \sum_{s \in \mathcal{S}_i} \sum_{\tau \in \mathcal{W}_s} \{ r_s h_{sv\tau} \} \tag{4.19}$$

The costs include five terms: (a) the purchase, development and manufacturing costs J_{pdm} of the commodities, modeled by Equation 4.20; (b) the launch costs J_{launch} of the commodities sent into space from Earth, modeled by Equation 4.21; (c) the penalty fees for delayed services J_{delay}, modeled by Equation 4.22; (d) the operating costs of the depots J_{depots}, modeled by Equation 4.23; and (e) the operating costs of the servicers J_{serv}, modeled by Equation 4.24.

$$J_{pdm} = \sum_{v \in \mathcal{V}} \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{N}_e} \sum_{j \in \mathcal{N}} \left\{ \sum_{k \in \mathcal{K}_C} c_k^{pdm} x_{vijtk}^{+} + \sum_{k \in \mathcal{K}_I} c_k^{pdm} m_k x_{vijtk}^{+} + c_v^{pdm} y_{vijt}^{+} \right\} \tag{4.20}$$

$$J_{launch} = c^{launch} \sum_{v \in \mathcal{V}} \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{N}_e} \sum_{j \in \mathcal{N}} \left\{ \sum_{k \in \mathcal{K}_C} x_{vijtk}^{+} + \sum_{k \in \mathcal{K}_I} m_k x_{vijtk}^{+} + m_v y_{vijt}^{+} \right\} \tag{4.21}$$

$$J_{delay} = \sum_{i \in \mathcal{N}_c} \sum_{s \in \mathcal{S}_i} \sum_{\tau \in \mathcal{W}_s} \left\{ c_s^{delay} \left(\tau - \tau_s \right) \sum_{v \in \mathcal{V}_{serv}} h_{sv\tau} \right\} \tag{4.22}$$

$$J_{depots} = \sum_{v \in \mathcal{V}_{depot}} \sum_{i \in \mathcal{N}_p} \sum_{t \in \mathcal{T}} \left\{ c_v^{depot} \Delta_{iit} y_{viit}^{+} \right\} \tag{4.23}$$

$$J_{serv} = \sum_{v \in \mathcal{V}_{serv}} \sum_{i \in \mathcal{N}_c \cup \mathcal{N}_p} \sum_{t \in \mathcal{T}} \left\{ c_v^{serv} \Delta_{ijt} y_{vijt}^{+} \right\} \tag{4.24}$$

In Equation 4.20 through Equation 4.24, c_k^{pdm} represents the purchase, development, and manufacturing costs per unit mass of non-vehicle commodities; c_v^{pdm} is the purchase, development, and manufacturing costs per vehicle commodity; c^{launch} is the launch cost per unit mass; c_s^{delay} is the cost incurred per unit time of delayed services; c_v^{depot} is the cost per unit time of operation of the depots; c_v^{serv} is the cost per unit time of operation of the servicers; m_k is the non-vehicle commodity mass; m_v is the mass vehicle commodity mass;

τ_s is the time at which service need s occurs; and Δijt is the duration of the arc between node i and node j at time t.

The profits to be maximized are then simply modeled by Equation 4.25.

$$J = J_{revenues} - J_{pdm} - J_{launch} - J_{delay} - J_{depots} - J_{serv} \tag{4.25}$$

4.2.3 Application of Rolling Horizon Decision Making to On-Orbit Servicing

For an OOS venture to make long-term and sustainable decisions early on, they must account for uncertainties in the servicing demand. In this chapter in particular, uncertainty stems from the randomly-occurring service needs (cf Table 3.2). Efficient methods such as the RH approach can be leveraged to support the decision making of OOS operators.

RH decision making is used to make decisions in a dynamic stochastic environment, typically characterized by uncertainties in demand and the resulting cost in forecasting this demand. More specifically, this technique consists in making optimal a series of decisions repeatedly based on short- to mid-term forecast of future demand. An advantage of RH is to decompose a complex dynamic scheduling problem into smaller sub-problems whose combined optimal solutions yield a satisfying solution of the original problem at a lesser computational cost. Readers are referred to Refs. [37] [38] for a more complete theoretical background on RH decision making.

The RH approach consists in dividing the total time horizon (called Scheduling Horizon (SH) in this chapter) in shorter time intervals (called Planning Horizons (PH)) over which the operations of a given system are optimized assuming an accurate demand forecast in those intervals. The results of the optimization over the first period of each PH (called Control Horizon (CH) in this chapter) are retained before “rolling on” the PH over time and reiterating on the described procedure. A critical step in the RH algorithm is to ensure the continuity of the state of the system between the end of a CH and the beginning of the following PH. Figure 4.6 concisely summarizes the traditional RH procedure with three successive PHs and CHs.

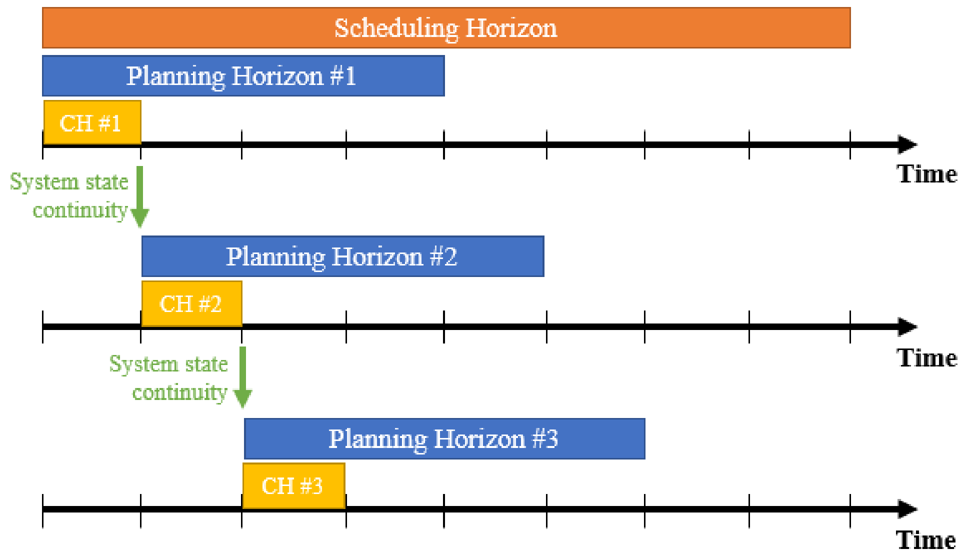

Figure 4.6: An illustration of the traditional Rolling Horizon procedure.

As depicted in Figure 4.6, the traditional RH approach re-plans the operations of the system every fixed time interval. This is possible when the uncertain demand can be forecast over the PH. In the context of OOS however, predicting the occurrences of random service needs such as satellite repairs is a difficult task and not appropriate for realistic operational scheduling. Therefore, a modification of the traditional RH procedure is proposed in which the OOS operations are re-planned whenever a random service need occurs. The OOS operations may also be re-planned on a regular basis if there has not been any random service need for a long time. The demand forecast over each PH is thus comprised of the random service need that triggers the re-planning of the operations (if any), and of the deterministic service needs arising over the PH. Finally, as for the traditional RH procedure, the state of the OOS infrastructure must be propagated from one CH to the next PH. Here, the state at any date t of the OOS infrastructure comprises the position and amount of all commodities in the static network at t.

Figure 4.7 through Figure 4.9 illustrate the proposed RH procedure in the context of OOS operational scheduling for a notional 6-node dynamic network. On those figures, the detailed time-expansion of the static network as presented in Figure 4.3 is omitted for readability. A notional service demand history is given to keep track over time of the service needs, their occurrence dates, and to which customer satellite the services must be provided. Figure 4.7 illustrates a notional schedule of a servicer over the first PH. The servicer's path over the dynamic network is represented by bold, yellow direct arcs. The operations are automatically re-planned after 120 days of operations as no random service need arises over that time interval (the 120-day time interval for automatic re-planning is user-defined). Figure 4.8 illustrates the servicer's schedule after the PH is rolled over between the dates t = 120 and t = 480. Note how the servicer's schedule from t = 0 to t = 120 is retained as part of the final solution whereas the servicer's schedule is modified after the date t = 120 due to a new deterministic service need, called d_5, in the forecast. At the date t = 200, a random service need called r_1 triggers a re-planning of the OOS operations. Thus, one can see that the CH in the proposed methodology does not have a fixed length contrary to the traditional RH procedure as shown in Figure 4.6. Finally,Figure 4.9 shows that the servicer's schedule is modified after the date t = 200 in order to provide a service in response to the random need r_1. Figure 4.7 through Figure 4.9 show the path of only one servicer for readability, but the proposed methodology is flexible enough to consider as many servicers as desired.

The OOS operations over each PH of the RH procedure are optimized by formulating the OOS logistics problem as a MILP model. Thus, the proposed methodology consists in solving a series of MILP problems over a given SH.

4.3 Application

This section demonstrates two use cases of the OOS logistics optimization framework: a short-term operational scheduling use case; and a long-term strategic planning use case. Subsection 4.3.1 summarizes the assumptions used to run the different scenarios designed to demonstrate the use cases. Subsection 4.3.2 demonstrates the proposed method as a tool

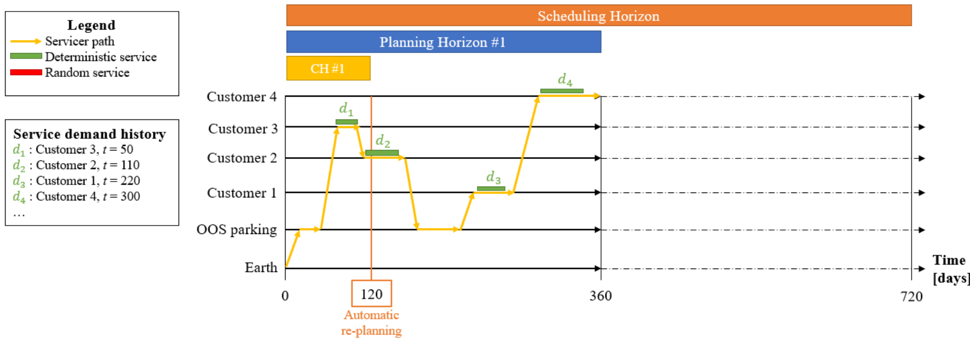

Figure 4.7: On-Orbit Servicing operations over the first planning and control horizons.

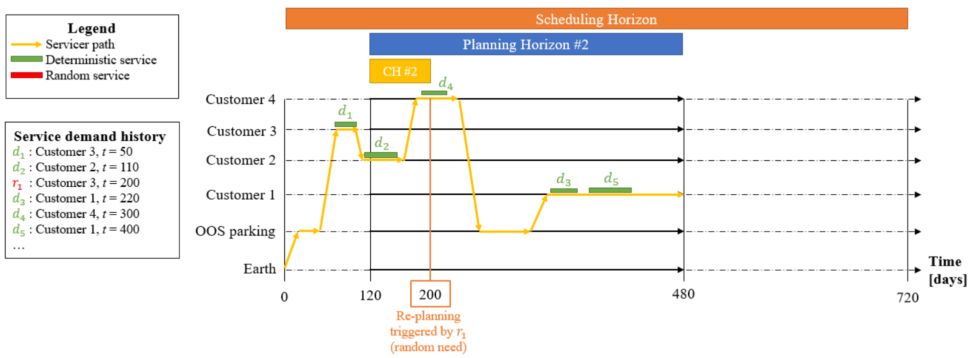

Figure 4.8: On-Orbit Servicing operations over the second planning and control horizons.

for short-term operational scheduling of an existing OOS infrastructure. Subsection 4.3.3 demonstrates the proposed method as a tool to inform long-term strategic design of OOS architectures by comparing the performance of different OOS architectures.

4.3.1 Assumptions and Scenarios

In this subsection are first presented the assumptions and data associated with the fleet of customer satellites. Then, the assumptions and data associated with the OOS infrastructure are given. Finally, the scenarios associated with the two use-cases are presented.

Customer Fleet Assumptions

The data related to the deterministic and random service needs are given in Table 4.6 and Table 4.7 respectively. Note that the given data are for one customer satellite; by increasing the number of customer satellites in the simulations, the service need rates of the entire

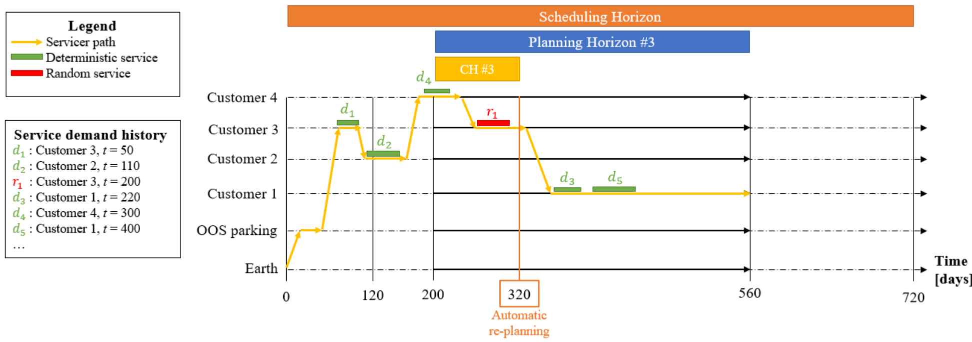

Figure 4.9: On-Orbit Servicing operations over the third planning and control horizons.

Table 4.6: Assumptions related to the deterministic service needs.

	Inspection	**Refueling**	**Station Keeping**
Revenues [$M]	10 [17]	15 [17]	20
Delay penalty fee [$/day]	5,000 [17]	100,000 [17]	100,000 [17]
Service duration [days]	10 [17]	30 [17]	180 [24]
Service window [days]	30	30	30
Frequency pf occurrence [days]	6,310 [21]	2,100 [21]	2,100 [21]

fleet of customer satellites increase. Also note that in Table 4.6 and Table 4.7, data without reference is assumed. The frequency of occurrence is derived from the data given in [21]. Finally, the positions of the customer satellites were retrieved from the UCS (Union of Concerned Scientists) database of satellites [40].

Table 4.7: Assumptions related to the random service needs.

	Repositioning	**Retirement**	**Repair**	**Mechanism Deployment**
Revenues [$M]	10 [17]	10 [17]	30	25 [17]
Delay penalty fee [$/day]	100,000 [17]	0 [17]	100,000 [17]	100,000 [17]
Service duration [days]	30	30	30	30
Service window [days]	30	30	30	30
Mean frequency of occurrence [days]	2,520 [21]	2,520 [21]	9,020 [21]	21,050 [21]

Table 4.8: Assumptions related to the servicers.

	Versatile (V)	Specialized 1 (S1)	Specialized 2 (S2)	Specialized 3 (S3)	Specialized 4 (S4)
Tools	T1, T2, T3, T4	T1	T2	T3	T4
Dry mass [kg]	3,000 [17]	2,000	2,000	2,000	2,000
Propellant capacity [kg]	1,000	1,000	1,000	1,000	1,000
Specific impulse [s]	316	316	316	30316	316
Manufacturing cost [$M]	75	50 [42]	50 [42]	50 [42]	50 [42]
Operating cost [$/day]	13,000 [25]	13,000 [25]	13,000 [25]	13,000 [25]	13,000 [25]

OOS Infrastructure Assumptions

The four notional servicer tools given in Table 3.5 each have an assumed cost of $100,000 and an assumed mass of 100 kg.

Two types of servicers are defined: one versatile servicer with four integrated tools; and four specialized servicers each with exactly one tool of each type. The versatile servicer can provide all seven defined services whereas the specialized servicers can only provide the services their tools are suited for. The detailed assumptions are given in Table 4.8. The baseline dry mass for the versatile servicer is taken from [17] but then reduced for the specialized servicers which are assumed to be less capable so of smaller size. The baseline propellant capacity in [17] is 500 kg but is increased in this chapter since this work models sustainable infrastructures. Thus, the servicers must be able to return to the depot to get refueled. The assumed specific impulse is characteristic of bipropellant rocket engine. Furthermore, this chapter assumes the refueling of the servicers to be instantaneous operations. The justification behind this assumption is that, as OOS operations become routine, refueling of the servicers will likely not take more than one time step (i.e., two days) [41]. This assumption can be modified depending on the technology performance.

An orbital depot is assumed pre-deployed at a GEO orbital slot located at a longitude of 170 deg West (over the Pacific Ocean). The depot is assumed to consume its own monopropellant at a rate of 0.14kg/day for station keeping [43]. The manufacturing and operating costs of the depot are assumed to be \$200M and \$13,000/day respectively.

The launch vehicle used for this analysis is based off a Falcon 9 launcher. One launcher is assumed to launch every 30 days to resupply the depot with a maximum payload capacity of 8,300 kg. The launch cost is assumed to be \$11,300/kg.

Finally, the commodities considered in the case studies are the spares (assumed price tag: \$1,000/kg), bipropellant for the servicers (price tag for monomethyl hydrazine: \$180/kg), and monopropellant (price tag for hydrazine: \$230/kg).

4.3.2 Simulation Scenarios

Using the assumptions presented previously, several scenarios are designed to demonstrate the short-term operational scheduling and long-term strategic planning capabilities of the proposed methodology. Three different levels of service demand are defined by considering 30 satellites (low demand rate), 71 satellites (medium demand rate), or 142 satellites (high demand rate).

The first case study consists in demonstrating the value of the proposed framework in optimizing short-term operational scheduling. This situation occurs when an OOS operator wishes to optimally re-plan the operations of its infrastructure for the next few months given a forecast of service needs. The OOS infrastructure is assumed to be initially pre-deployed at some customer satellites to account for the fact that at re-planning some servicers may already be providing services. Note that the proposed framework allows users to define any initial state of the OOS infrastructure. For the purpose of this demonstration, the framework is run for a single PH optimization. This scenario is summarized in Table 4.9.

The second case study demonstrates the value of the proposed framework in analyzing the pros and cons of a particular OOS architecture over the long term and trading different OOS architectures with one another. In order to assess the value of each OOS architecture, the optimization framework is run for a scheduling horizon of five years. Each planning horizon optimization is performed over 100 days and the length of the control horizon

Table 4.9: Scenarios definition for the use case of short-term operational scheduling

Servicers	4 specialized servicers (S1, S2, S3, S4)
Depot	1 depot pre-deployed at 170 deg West
Planning horizon	100 days
Customer fleet	142 customer satellites (high demand)
Initial conditions	- S1 pre-deployed at satellite UFO-4 profiving refueling service - S2 pre-deployed at satellite UFO-4 (due to prior service) - S3 pre-deployed at satellite Echostar 23 (due to prior service) - S4 pre-deployed at satellite Galaxy 28 providing station keeping service

Table 4.10: Scenarios definition for the use case of short-term operational scheduling

	Monolithic architecture	**Distributed architecture**
Servicers	1 versatile	4 specialized
Depot	1 depot pre-deployed at 170deg West	1 depot pre-deployed at 170deg West
Planning horizon	100 days	100 days
Control horizon (in case of automatic re-planning)	60 days	60 days
Initial conditions	The servicer is pre-deployed at the OOS parking node	All servicers are pre-deployed at the OOS parking orbits

in case of automatic re-planning is 60 days. For this case study, two different OOS architectures are considered: a monolithic architecture using one versatile servicer; and a distributed architecture using four specialized servicers. The value and profitability of each architecture is assessed over five years of operations under low service demand (30 satellites), medium service demand (71 satellites) and high service demand (142 satellites). The value of an OOS architecture is defined as the difference between the profits and the initial investments. Thus, an OOS architecture is considered valuable at a given date if its value is positive. Table 4.10 summarizes the parameters related to each OOS architecture.

4.3.3 Use Case 1: Short-Term Operational Scheduling

Assume that an OOS company owns a fleet of vehicles to provide orbital services to a set of satellites all located in GEO. Whenever it decides to re-plan its operations, the OOS optimization framework would be run for a few hours (depending on the problem size) and get an output of the flow of operations. Note that not all customer satellites will need a service during the period considered for re-planning. The framework thus includes in the network only the satellites that experience service needs over the PH. This minimizes the size of the optimization problem thus allowing for a faster solving. For example, in the scenario presently considered, the customer base is of 142 satellites. However, over the considered PH, only 10 satellites are included in the network as the other satellites do not experience any service need during that period. Of these 10 satellites, only eight are visited in the final solution. Figure 4.10 gives the network for the considered scenario. The customer satellites are represented by colored dots. The red dots correspond to the visited customer satellites, whereas the blue dots are not visited by any servicer.

The optimization framework is run on an Intel® Core™ i7-9700, 3.00GHz platform with the Gurobi 9 optimizer. The solution was reached in less than 10 seconds with a gap of 1% as stopping criterion. For this case study, the time-expansion parameters are Δt = 2 days, T = 10 days, n = 2. The outputs of the optimization are depicted on Figure 4.11 and Figure 4.12.

It is worth noting on Figure 4.11 and Figure 4.12 that the framework optimally leverages the service windows defined to give the OOS planners more flexibility in assigning the services to the servicers. For example, the S2 servicer is assigned the "Inspection" service at the Echostar 12 satellite before that at the UFO-2 satellite whereas the service need at UFO-2 arises first. The optimizer decides to plan this way the operations of servicer S2 because Echostar 12 is "on the way" between the depot and UFO-2. Doing so, servicer S4 saves propellant and thus limits the cost associated with its operations. The same observation can be made between the "Refueling' services provided at Echostar 14

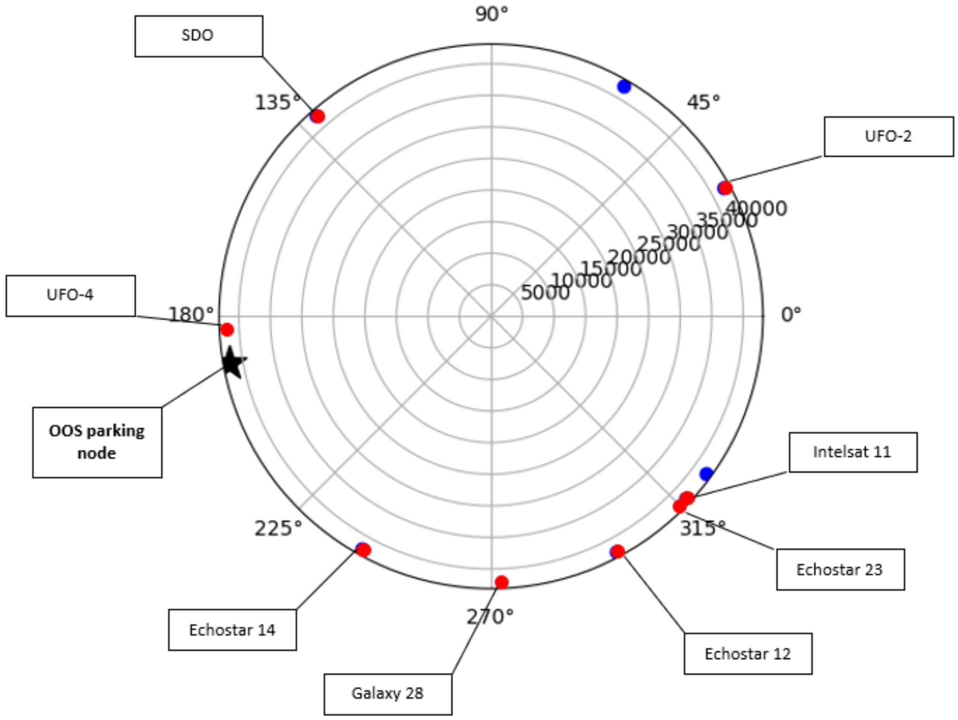

Figure 4.10: Static network used to demonstrate the short-term operational scheduling of the OOS infrastructure.

and SDO.

Also noticeable is the return of servicer S1 to the depot at times 22 and 92. The servicer returns to the depot and then immediately departs for other services. This is because the servicer needs to be refueled by the depot, which is assumed to be instantaneous as mentioned earlier.

For information, the optimal operations presented on Figure 4.11 and Figure 4.12 generate based on the initial assumptions \$65M of revenues for a total operating cost of \$46.5M. As depicted on the figures, the operations of the OOS infrastructure are profitable. These values however may not be final for the period $t = 0..100$ as re-planning may be needed at a later time due to a randomly occurring service need. When a random need occurs, the OOS planners would re-run the framework with the initial conditions corresponding to the current orbital state of the OOS infrastructure and an updated set of service needs.

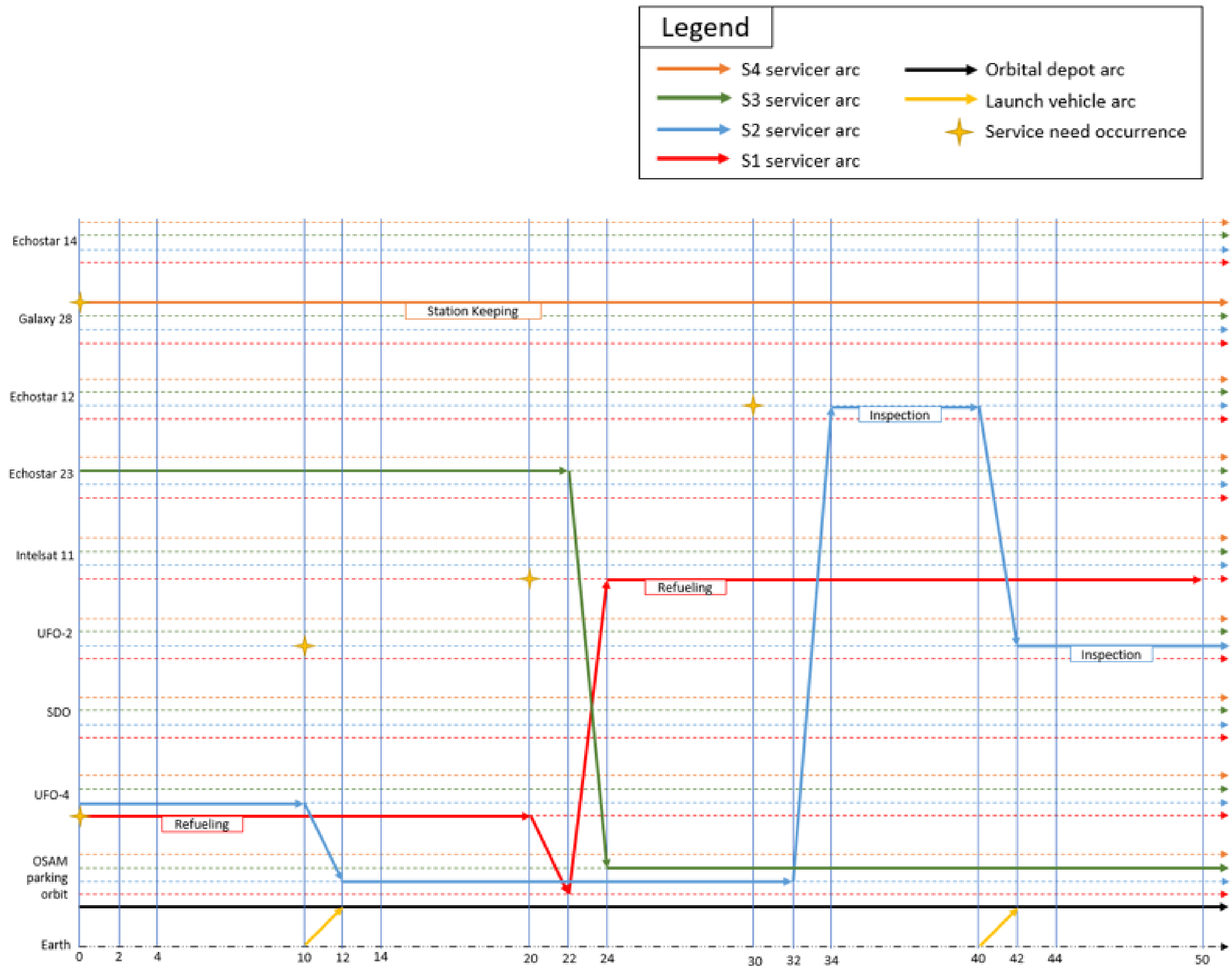

Figure 4.11: Breakdown over the first 50 days of the operational scheduling of a notional OOS infrastructure consisting of four specialized servicers and a depot (Δt = 2 days, T = 10 days, n = 2).

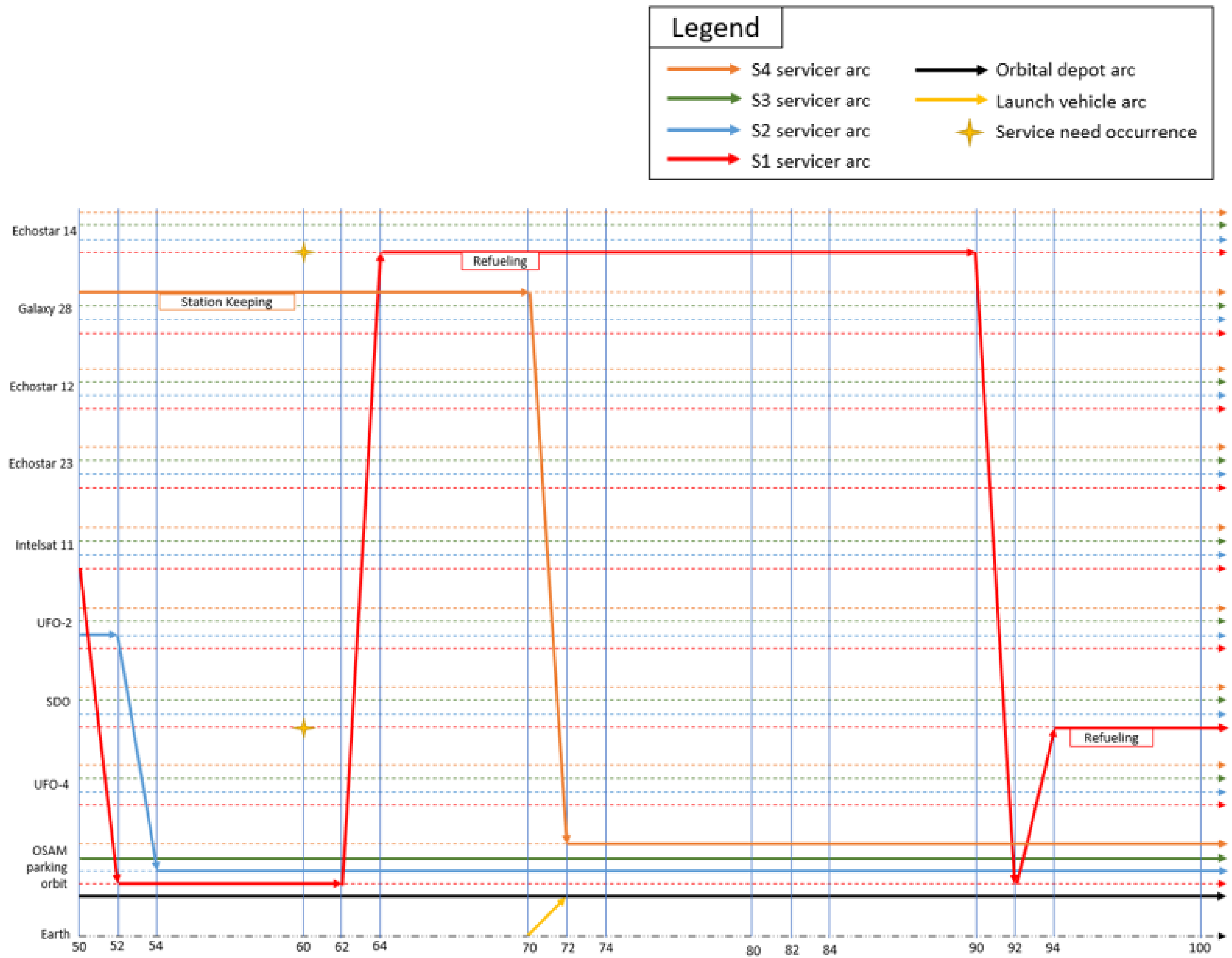

Figure 4.12: Breakdown over the last 50 days of the operational scheduling of a notional OOS infrastructure consisting of four specialized servicers and a depot (Δt = 2 days, T = 10 days, n = 2).

4.3.4 Use Case 2: Long-Term Strategic Planning

Besides being used as a short-term operational scheduling tool, the optimization framework may also be leveraged as a long-term strategic planning tool to assess the advantage of a given OOS architecture over another one and/or assess the impact of exogenous variables (i.e., non-controllable by the OOS planners) such as the state of the on-orbit servicing market (e.g., service demand level). In this chapter, the OOS architectures are compared based on the profits they generate and their values at the end of the scheduling horizon. This case therefore demonstrates the developed method as a tool to design the architecture for OOS considering a long time horizon.

Additional aspects need to be considered for the long-term strategic planning. To plan the operations of an OOS infrastructure over the short-term, only deterministic service needs are considered. These include for instance the services resulting from a contract between the OOS operator and a GEO satellite operator who wants its satellites to be regularly refueled. In the long-term strategic planning context, the uncertainty associated with services that cannot be planned in advance (e.g., a repair) is captured through the RH approach. Two types of service needs are random by nature: repair and mechanism deployment. To some extent, retirement and repositioning service needs may also be considered random, as assumed here.

As a demonstration example, the value and profitability of a distributed architecture (i.e., four small servicers carrying a different tool) are compared to that of a monolithic architecture (i.e., one large servicer carrying all tools needed to provide any service). OOS planners may also want to see the impact of the market on the metrics for each architecture. For example, a costly architecture such as one involving many small, specialized servicers may not be as profitable as a monolithic one if the demand is low. But what if the market is expected to significantly grow in the near term? Wouldn't a distributed architecture be more fit to capture the extra sources of revenue and thus outweigh its initial investment at a faster rate? Those are questions that the framework can help answer.

In order to demonstrate the value of the framework for long-term strategic planning, it is run for the two architectures outlined in Table 13 over a scheduling horizon of five years and for three levels of service demand (30, 71 and 142 satellites). The successive optimizations were solved on an Intel® Core™ i7-9700, 3.00GHz platform with the Gurobi 9 optimizer. The two stopping criteria used in the optimizations were a gap of 1% and a time limit of 7200s. In total, 104 successive optimizations were solved in an average time of 690s and with an average gap of 1.22%. Four optimizations reached the 7200s time limit before reaching the required gap; in which case the best feasible solutions are used. As for the previous case study, the time-expansion parameters are Δt = 2 days, T = 10 days, and n = 2.

The customer base of the OOS infrastructure is assumed to be of 30 satellites. Figure 4.13 compares the evolution of the values of the monolithic and distributed architectures. Figure 4.14 and Figure 4.15 respectively represent the revenues generated and costs incurred by both infrastructures over time. The architectures are equivalent in terms of profitability as shown by the slopes of the value curves. In this case, the single servicer in the monolithic architecture is not overwhelmed by too large a number of service needs and can thus provide almost as many services as the distributed architecture with an overall lower cost. Note that based on the assumptions summarized in Table 4.8, the initial investment needed for the distributed architecture is significantly larger. This would need to be accounted for by decision makers regarding the design of an OOS infrastructure.

It is now assumed that the customer base more than doubles (71 satellites). Figure 4.16 represents the values of both architectures. Figure 4.17 and Figure 4.18 respectively represent the revenues generated and costs incurred by both infrastructures over time. It is now clear that the distributed architecture outpaces the monolithic architecture in terms of profitability. It even becomes more valuable after three years of operation. In order to put Figure 4.13 and Figure 4.16 in perspective, assume that the forecast for the first five years is on the order of 30 satellites and that the following five years, the customer base doubles.

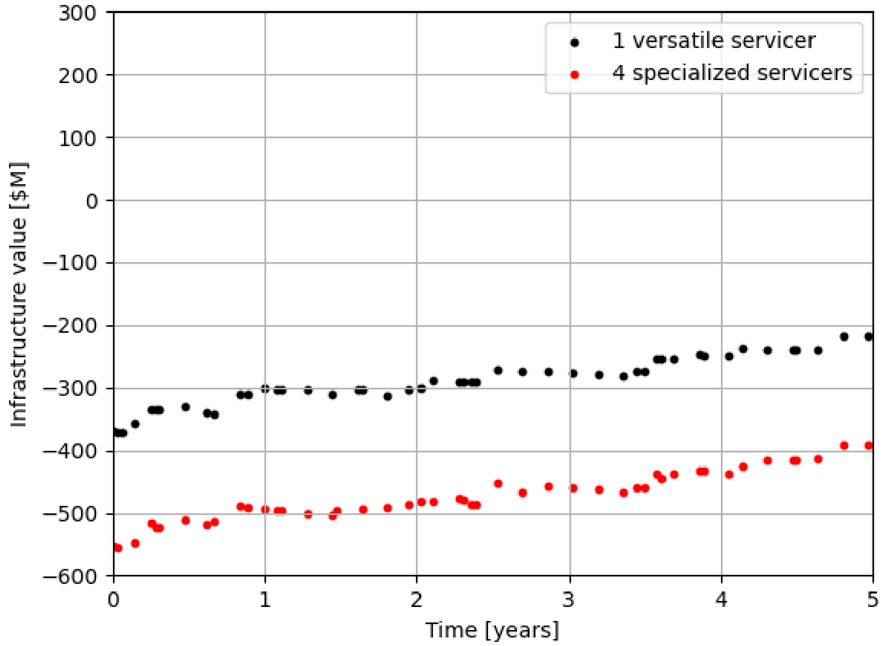

Figure 4.13: Values of the monolithic and distributed architectures for a customer base of 30 satellites.

Two of the decisions OOS planners may face are:

- wait until the market becomes more favorable so as to avoid a long payback period; or
- invest early in a distributed architecture whose profitability after a few years of operations would increase due to a growing customer base.

Investing early, the OOS operator may gain a leading competitive edge against new entrants as the market grows. This would also mean convincing investors to endure a longer payback period.

Finally, it is assumed that over the years the confidence in orbital servicing grows significantly. Figure 4.19 simulates the values of both infrastructures for a customer base of 142 satellites. Figure 4.20 and Figure 4.21 respectively represent the revenues generated and costs incurred by both infrastructures over time. Unlike the trends observed in Fig-

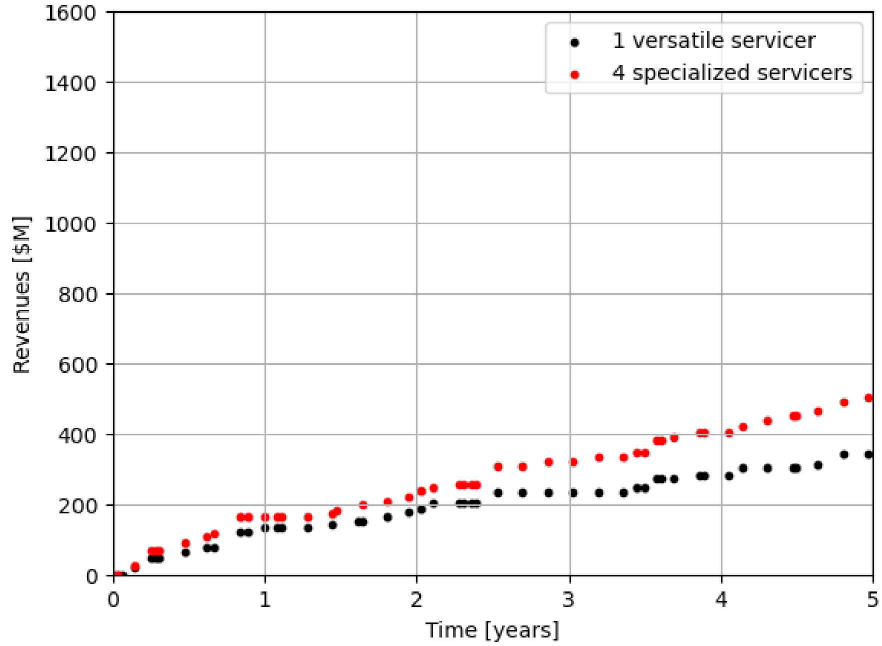

Figure 4.14: Revenues generated by the monolithic and distributed architectures for a customer base of 30 satellites.

ure 4.16, the OOS infrastructures can pay for themselves after 2.5 years of operation for the distributed architecture and after five years for the monolithic architecture. However, as observed previously, the distributed architecture has a leading competitive edge over the monolithic architecture, being much more profitable.

Rather than providing definitive conclusions about the best OOS architectures given some market conditions, this section aims to demonstrate that the framework can handle the modeling and analysis of a wide spectrum of potentially complex infrastructures. A complex infrastructure was chosen to show the capability of the methodology, but the same approach can be used for a simpler, near-term infrastructure case as well; the generality is an advantage of the proposed approach

The current framework can also easily be modified to model a dynamic evolution of the servicing demand. For example, as the confidence in OOS improves, the market size may progressively increase. This would mean that precursor OOS infrastructures could

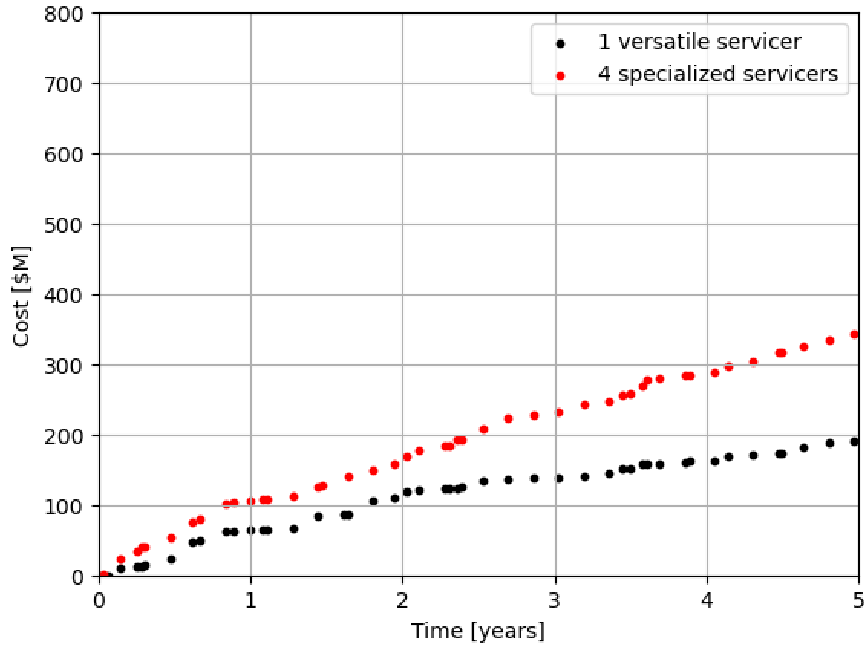

Figure 4.15: Costs incurred by the monolithic and distributed architectures for a customer base of 30 satellites.

lose market shares if they reach their maximum servicing capacities. Given some assumed dynamic market forecast, sensitivity analyses could be run using the framework to gain insights into how and when to update an existing OOS infrastructure.

4.4 Summary

This chapter develops an optimization framework to enable on-orbit servicing operators to make rigorous decisions regarding the short-term operations and/or the long-term strategic planning of their fleets of servicers and orbital depots. To this end, three innovations are achieved: (1) model the operations of sustainable many-to-many OOS infrastructures as a logistics network; (2) extend the traditional space logistics MILP formulation with additional variables and constraints that model realistic OOS operations; and (3) account for the uncertainties in service demand by adapting the RH decision making approach to OOS logistics.

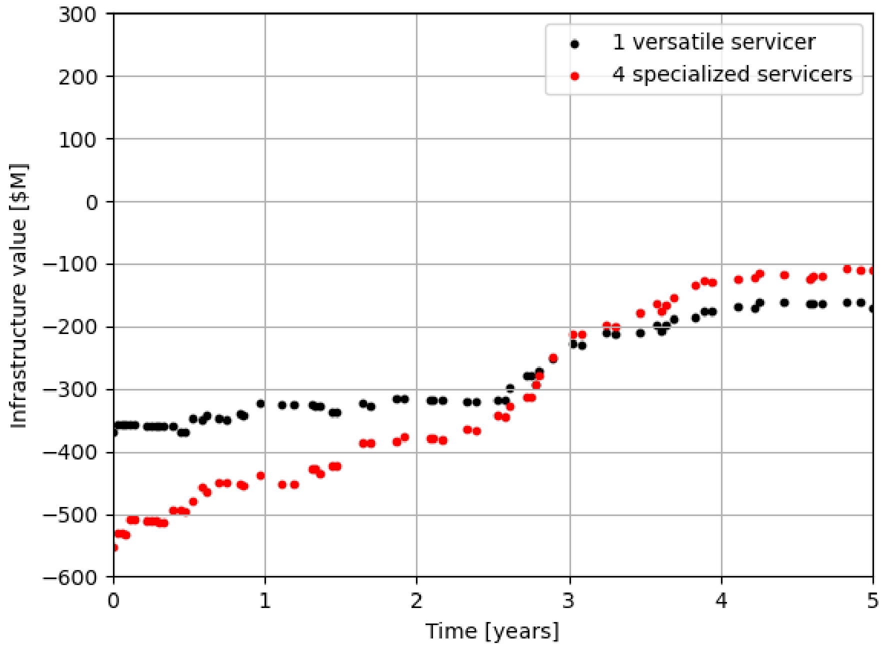

Figure 4.16: Values of the monolithic and distributed architectures for a customer base of 71 satellites.

The proposed framework is developed based on three main assumptions: (1) the fleet of customer satellites is distributed along a shared circular orbit; (2) the servicers use high-thrust technology to perform orbital maneuvers; and (3) the orbital depots, if deployed, are located on the same circular orbit as the fleet of customer satellites.

Two different case studies are explored to demonstrate the framework's value in the context of GEO servicing. The first case study shows the framework's ability in efficiently optimizing short-term operational scheduling of existing OOS infrastructures. This is done by running it over a single PH. The second case study proves the framework's value in exploring the design tradespace of OOS architectures under various OOS market conditions. This provides answers as to what features an OOS architecture should exhibit to draw the most benefits out of an assumed customer base. In this chapter, three five-year simulations are performed with three different service demand rates in order to compare two fundamentally different OOS architectures: a monolithic architecture that uses a single versatile

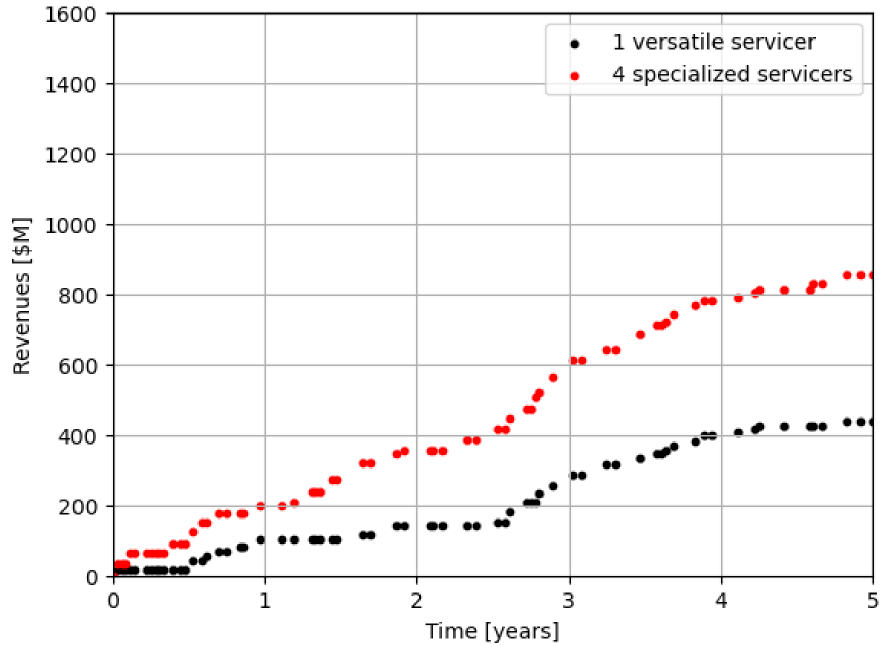

Figure 4.17: Revenues generated by the monolithic and distributed architectures for a customer base of 71 satellites.

servicer versus a distributed architecture that leverages four specialized servicers.

To conclude, the framework proposed in this chapter uniquely extends the network-based space logistics technique and the RH approach, and combine them to the OOS design and operational optimization under uncertain demands. Future work will relax the assumptions made in this chapter by modeling the low-thrust propulsion technology and trajectories of the servicers and by considering customer fleets distributed over different orbits.

This chapter is based on the following publication:

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, 2021

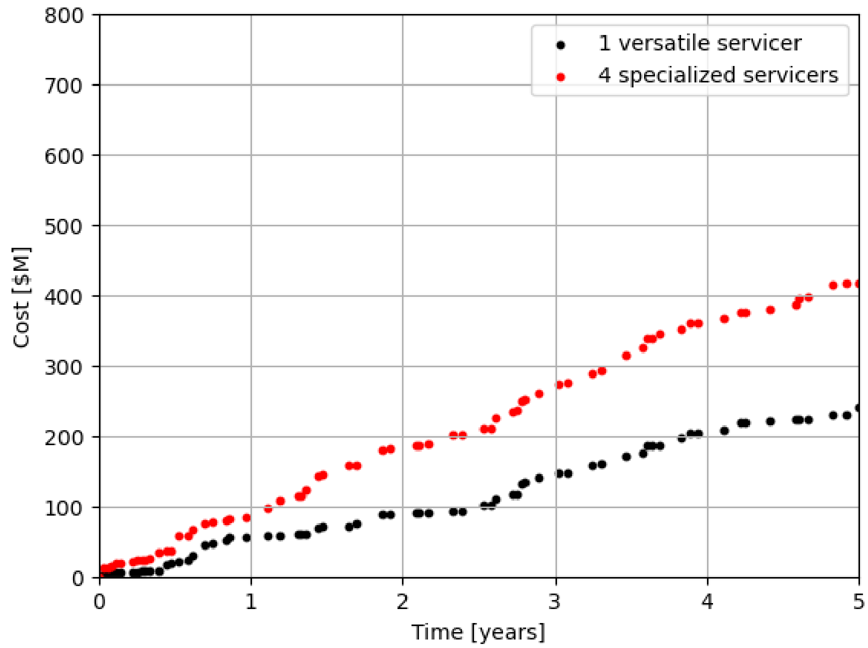

Figure 4.18: Costs incurred by the monolithic and distributed architectures for a customer base of 71 satellites.

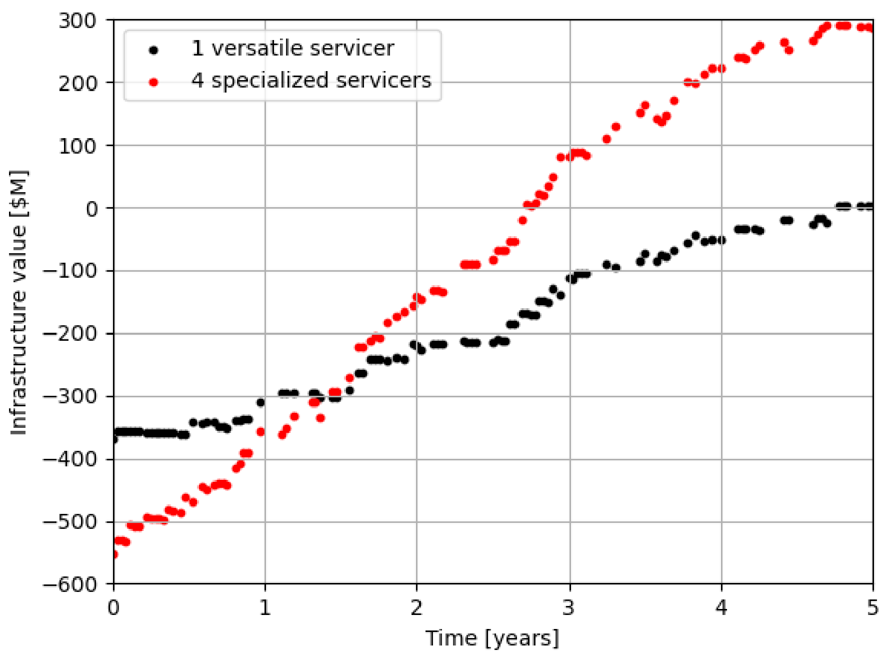

Figure 4.19: Values of the monolithic and distributed architectures for a customer base of 142 satellites.

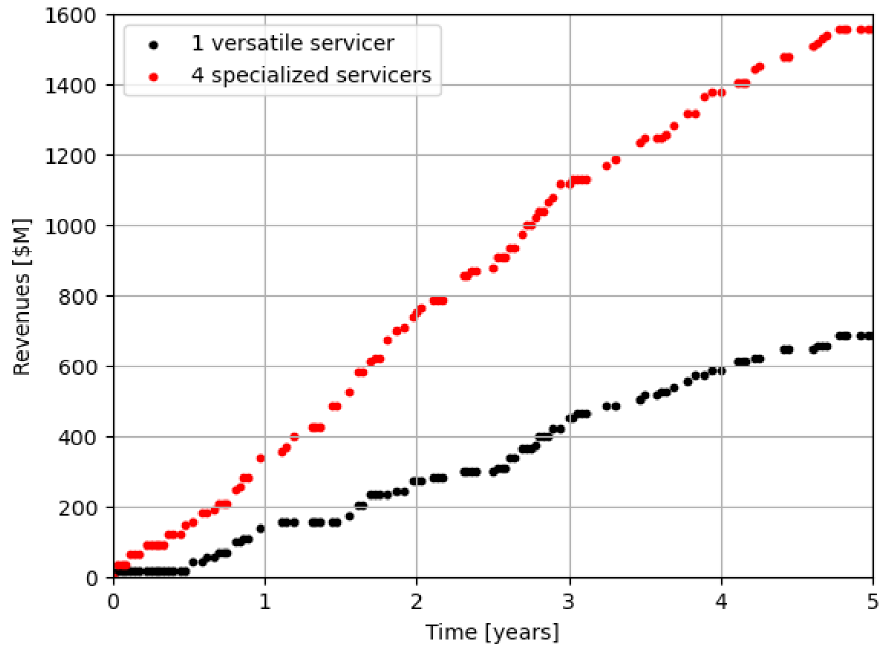

Figure 4.20: Revenues generated by the monolithic and distributed architectures for a customer base of 142 satellites.

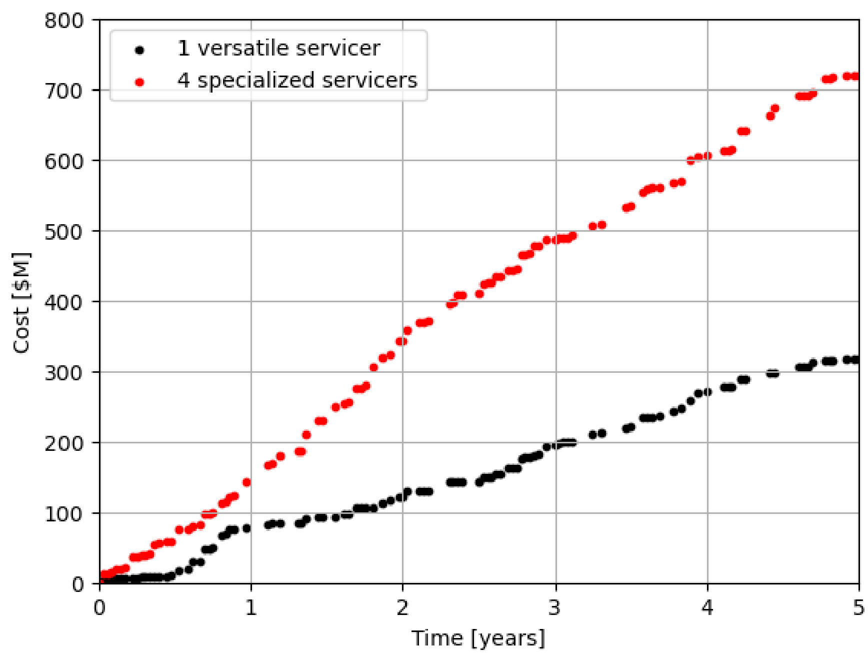

Figure 4.21: Costs incurred by the monolithic and distributed architectures for a customer base of 142 satellites.

CHAPTER 5
ON-ORBIT SERVICING OPTIMIZATION FRAMEWORK WITH HIGH- AND LOW-THRUST PROPULSION TRADEOFF

5.1 Scope

This chapter generalizes the OOS logistics method developed in chapter 4 by enabling the automated tradeoff between high-thrust, low-thrust, and multimodal servicers (i.e., servicers with both high-thrust and low-thrust propulsion capabilities). This is achieved by (1) combining the concepts of dynamic generalized multi-commodity network and piecewise linear approximation for the non-linear low-thrust trajectory model; and (2) leveraging the concept of *multiarcs* to effectively give the optimizer discrete user-defined options related to the propulsion systems and trajectories of the servicers.

The orbital depots and customer satellites, as in chapter 4, are co-located along the same circular orbit, as depicted in Figure 5.1.

5.2 On-Orbit Servicing Logistics Formulation

In this section, the OOS logistics formulation presented in chapter 4 is generalized by enabling the modeling of high-thrust, low-thrust, and multimodal servicers. Subsection 5.2.1 presents the dynamic network used to represent the operations of the OOS infrastructures and the flights of the servicers. Subsection 5.2.2 describes how the servicers' trajectory models are computed by user-defined plugins interfacing with the proposed OOS logistics framework. Subsection 5.2.3 gives the new MILP model which can capture the tradeoff between high- and low-thrust propulsion technologies.

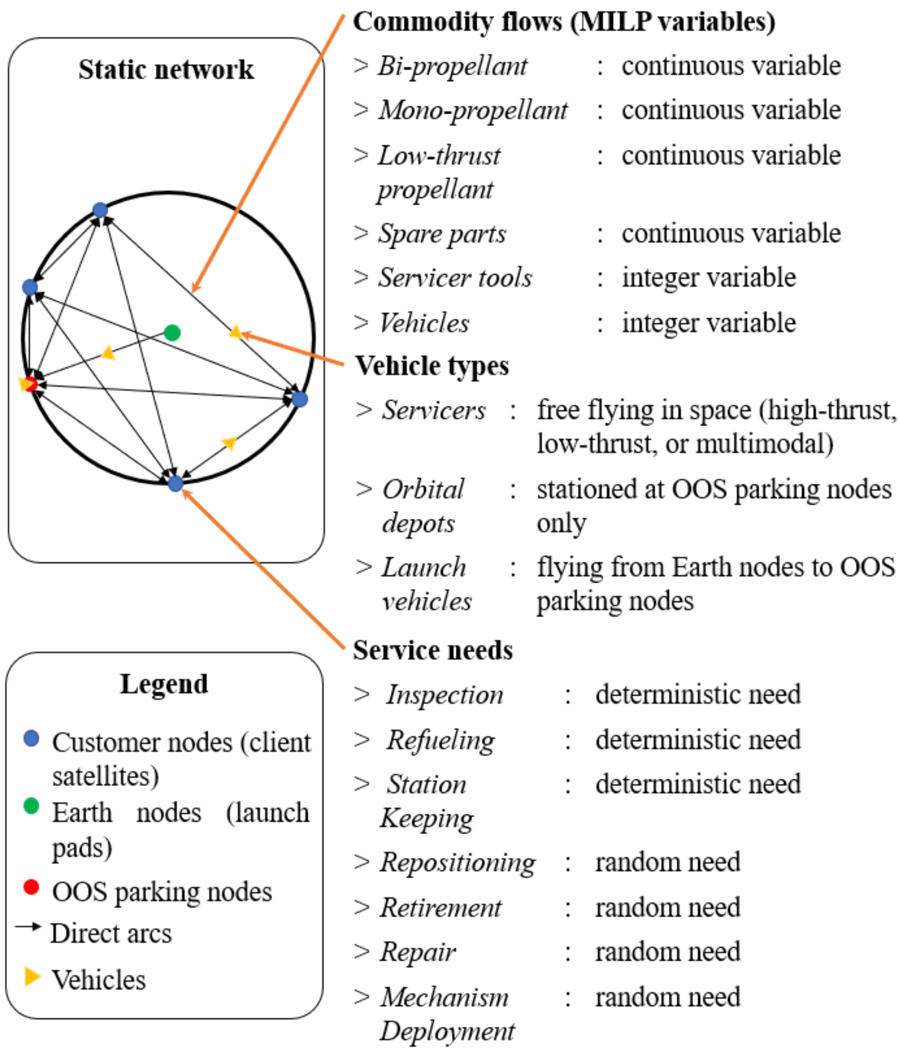

Figure 5.1: High-level overview of the static OOS logistics network; modified from chapter 4.

5.2.1 Dynamic network

The static network given in Figure 5.2 is expanded at predefined time steps by following a periodic topology as illustrated in Figure 5.2 with a simple three-node network. The period of the network, denoted T, is chosen to be at most as large as the greatest common divisor of the durations of the considered services. The network is replicated at every period T and at two additional time steps within each period. This essentially creates three intervals per period T. The first two are used to enable short spaceflights such as that of the high-thrust servicers and launch vehicles. The third one is used for the provision of services. Figure 5.2 illustrates the route of a high-thrust servicer in bold yellow arrows. It is sent to orbit by a launch vehicle between $t = 0$ and $t = 2$ before performing on its own an orbital maneuver to rendezvous with a customer satellite between $t = 2$ and $t = 4$. Once the servicer's task is completed, the servicer goes back to its parking location on a four-day trajectory between $t = 10$ and $t = 14$.

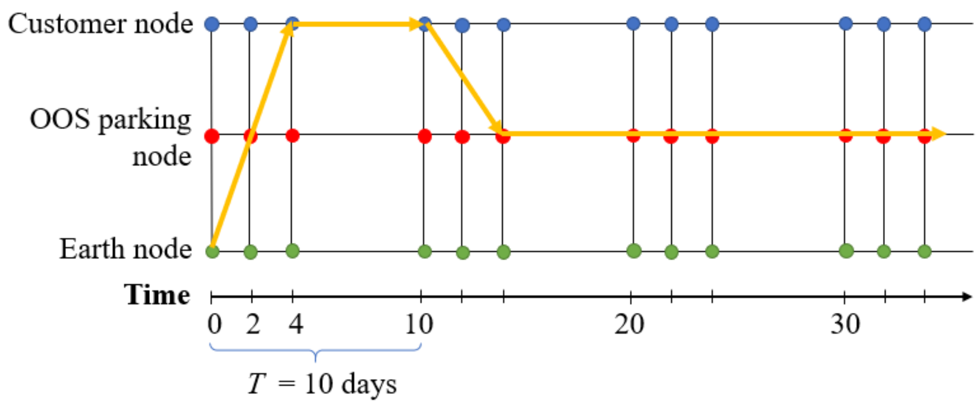

Figure 5.2: Notional dynamic network and high-thrust servicer path; modified from Chapter 4.

In order to model the long flight durations characteristic of low-thrust servicers and trajectories, the transportation arcs are allowed to span more than the first two intervals of each period T. This is depicted in Figure 5.3 with the notional path of a low-thrust servicer. The servicer is sent to orbit on a launch vehicle between $t = 0$ and $t = 2$ before performing a 12-day rendezvous maneuver to reach a customer satellite. Once its task is complete, it comes back to its parking location on a 14-day trajectory between $t = 20$ and $t = 34$. Note that the transportation arcs introduced in Figure 5.2 and Figure 5.3 could in fact be multiarcs representing different strategies for the rendezvous maneuvers performed by the servicers.

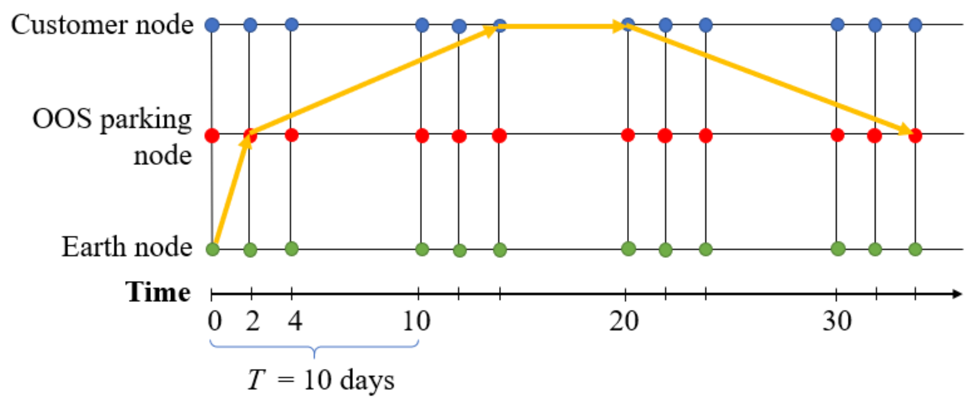

Figure 5.3: Notional path of a low-thrust servicer.

5.2.2 Servicer flight options and trajectory plugins

In order to give several servicer flight options to the optimizer, the framework users can define as many flight durations as desired. This allows the optimizer to automatically trade the propellant consumption of the servicers (i.e., cost) and their time of flight (i.e., responsiveness). This is illustrated in Figure 5.3 where the servicer first flies from the OOS Parking Node to the Customer Node with a 12-day rendezvous maneuver, and then back to the OOS Parking Node on a 14-day trajectory.

Similarly, the optimizer can be given multiple trajectory options for the flights of the servicers. For example, a high-thrust servicer could use a two-impulse maneuver or a three-impulse maneuver to rendezvous with a customer satellite. The optimizer would then choose which trajectory is better suited to maximize the profits generated by the overall OOS infrastructure. The trajectory options given to the optimizer are modeled in user-defined trajectory plugins interfacing with the OOS logistics framework through standard inputs and outputs described in Table 5.1. Each plugin first computes a mathematical model of the propellant consumption of a servicer as a function of its initial mass. This model is computed over a finite range of the initial mass of the servicer due to the varying masses of propellant and spares in the servicers' tanks and cargo bays. If the model is non-linear – often true of low-thrust models – it cannot be used as is in a MILP. In this case, the plugin automatically converts the non-linear model into a piecewise linear model and is incorporated into the MILP formulation as introduced in the next subsection. In addition, the plugin computes the initial mass of the servicer beyond which the trajectory becomes infeasible (referred to as *servicer mass upper bound* in this chapter). The servicer mass upper bound is critical to ensure that overloaded servicers cannot fly over infeasible trajectories (cf Equation 5.26).

A typical trajectory plugin workflow is summarized in Figure 5.4. The high-thrust and low-thrust trajectory models described in section 1.8 are each integrated into a plugin interfacing with the OOS logistics framework. As many trajectory plugins can be defined and interfaced with the OOS logistics framework as long as they accept the standard inputs and yield the standard outputs given in Table 5.1.

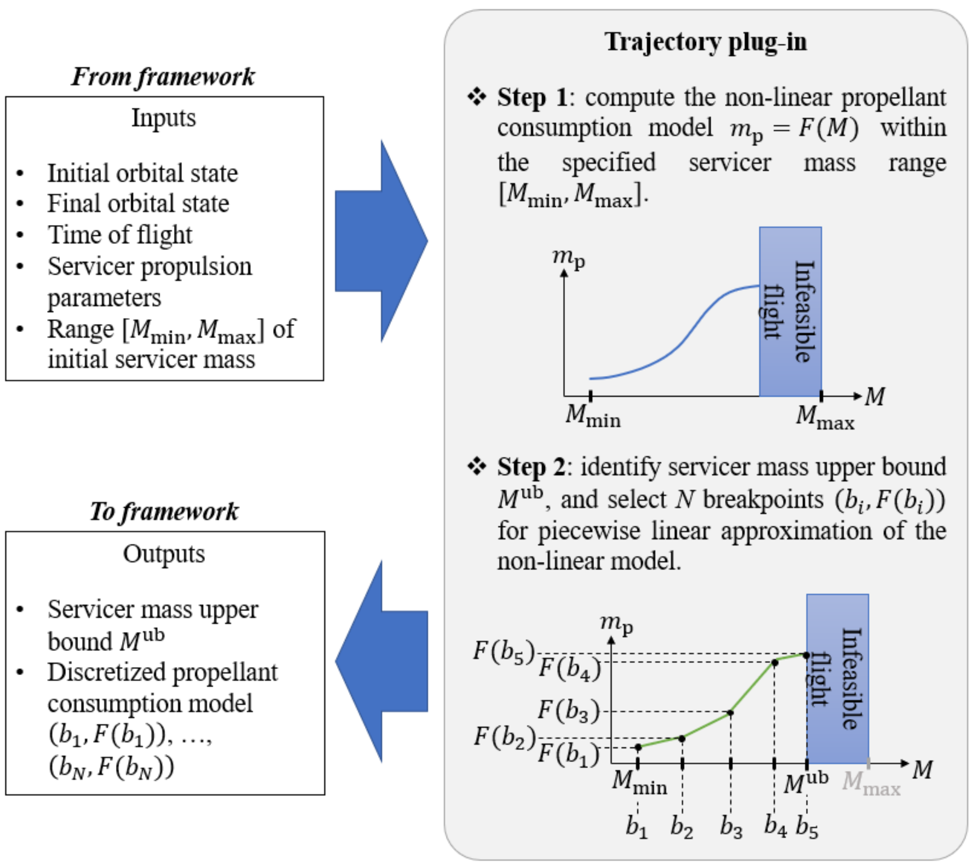

Figure 5.4: Typical trajectory plugin workflow and interfacing with OOS logistics framework.

Table 5.1: Definitions of the trajectory plugins' standardized input and output parameters.

	Parameter	Definition
Inputs	*Initial orbital state*	Initial position of the servicer in the network at the beginning of the rendezvous maneuver.
	Final orbital state	Final position of the servicer in the network at the end of the rendezvous maneuver.
	Time of flight	Duration of the rendezvous maneuver.
	Servicer propulsion parameters	Parameters needed to compute the propellant consumption model, such as the specific impulse and the thrust of the engine.
	Range of initial servicer mass	Bounds of the mass of the servicer within which the propellant consumption model is computed.
Outputs	*Servicer mass upper bound*	Initial mass of the servicer beyond which the trajectory becomes infeasible. This may happen in the case of a low-thrust servicer trajectory if the specified time of flight is too short.
	Discretized propellant consumption model	Breakpoints selected from the computed non-linear propellant consumption model for piecewise linear approximation and integration within a MILP. A breakpoint corresponds to an abscissa-ordinate pair, with the abscissa being the initial mass of the servicer and the ordinate being the associated propellant mass consumed.

5.2.3 Mathematical formulation

This subsection gives the complete MILP model used to optimize the operations of sustainable OOS infrastructures. This model is built upon the dynamic network presented in 5.2.1. The OOS logistics MILP formulation presented in chapter 4 is generalized in three ways:

1. New index sets and variables are introduced to model the multiarcs associated with the various propulsion system and trajectory options of the servicers;

2. Additional constraints are defined to properly model the low-thrust rendezvous maneuvers of low-thrust and multimodal servicers (cf Equation 5.17, Equation 5.18, Equation 5.19, Equation 5.26);

3. Constraints are added to model realistic servicer operations, whether they be related to the provision of services (cf Equation 5.24) or to the flights of the servicers (cf Equation 5.27).

After the model is formulated, it is solved either once to optimize the short-term scheduling of the OOS operations, or many times in a row, using the RH procedure, to assess the value of a candidate OOS architecture over the long term. RH decision making is a business practice commonly used in dynamic stochastic settings by which the most immediate decisions are made based on a forecast of near- to medium-term relevant information (e.g., demand for a product or service).

5.2.3.1. Index sets and variables

Consider a time-expanded network graph $\mathcal{G}$ comprised of a set of nodes $\mathcal{N}$ and a set of directed arcs drawn between any two nodes. The set of nodes $\mathcal{N}$ is the union of three subsets: the set $\mathcal{E}$ of Earth Nodes; the set $\mathcal{P}$ of OOS Parking Nodes; and the set $\mathcal{C}$ of Customer Nodes. For convenience in writing the objective function and constraints, a set of Orbital Nodes is defined as $\mathcal{N}_{\text{orb}} = \mathcal{P} \cup \mathcal{C}$. The nodes of the network are indexed by the letters i and j.

There are two types of directed arcs in the network: holdover and transportation arcs. A holdover arc connects a node with itself at two successive time steps. A transportation arc connects two different nodes at two different, not necessarily successive, time steps. Examples of holdover and transportation arcs are illustrated in Figure 5.2 and Figure 5.3 as thin blue lines and bold yellow arrows respectively. Holdover arcs are characterized by three indices: the type of vehicle v flying along the arc, a node i which is both the start and end node of the arc, and the start time t of the arc. Transportation arcs are characterized by six indices: the type of vehicle v flying along the arc, the start node i, the end node $j \neq i$, the time of flight q of the rendezvous maneuver, the type of trajectory r, and the start time t of the arc. Note that the indices q and r generalize the OOS logistics MILP formulation given in chapter 4 by enabling multiarcs that represent various propulsion and trajectory options.

The vehicle types v are defined in index set $\mathcal{V}$. As subsets of $\mathcal{V}$ are $\mathcal{V}_{\text{serv}}$ and $\mathcal{V}_{\text{dep}}$ which encompass the different servicer designs and depot designs, respectively. A set $\mathcal{Q}_v$ of time of flight options (index: q) is defined per vehicle type v. This allows the users of the framework to define a different set of flight duration values per servicer design. Similarly, the set $\mathcal{R}_{vq}$ (index: r) stores the trajectory options per vehicle type v and time of flight q. The index r thus refers to the trajectory models defined in the user-defined plugins. Finally, the set of time steps t is denoted by $\mathcal{T}$.

The commodities are represented with the index k taking its values in the set $\mathcal{K}$. Subsets of $\mathcal{K}$ are the set $\mathcal{V}_{\text{cont}}$ of continuous commodities (e.g., propellant), and the set $\mathcal{K}_{\text{int}}$ of integer commodities. A subset of integer commodities is the set of servicer tools denoted with $\mathcal{K}_{\text{tools}}$. The flow of commodities along the arc is captured by four sets of variables. The variables $X^{\pm}_{vitk}$ represent the node outflow (superscript $+$) and node inflow (superscript $-$) of commodity k along holdover arcs. They are continuous or integer nonnegative variables based on the nature of commodity k. Similarly, the binary variables $Y^{\pm}_{vit}$ represent the node outflow and inflow of the vehicle commodity along holdover arcs. The variables $U^{\pm}_{vijqrtk}$ represent the node outflow and inflow of commodity k along transportation arcs. They are continuous or integer nonnegative variables based on the nature of commodity k. Finally, the binary variables $W^{\pm}_{vijqrt}$ represent the node outflow and inflow of the vehicle commodity along transportation arcs. For conciseness in certain constraints, the non-vehicle commodity variables are represented under their column vector form as $\boldsymbol{X}^{\pm}_{vit}$ and $\boldsymbol{U}^{\pm}_{vijqrtk}$ along holdover arcs and transportation arcs, respectively.

Lastly, the management of services is captured through two additional sets of binary variables. The first one is the set of *service assignment* variables H_{vst} which specify at what time step t the servicer $v \in \mathcal{V}_{\text{serv}}$ must start addressing service need s. The service need is addressed if $H_{vst} = 1$. These variables are defined over a subset of $\mathcal{T}$, denoted $\mathcal{W}_s$ (indexed by service need s), that encompasses the optional dates to start addressing service need s. $\mathcal{W}_s$ is referred to as the *service window* within which the service must start if the optimizer decides to provide it.

The second set of service management variables is the set of *servicer dispatch* variables B_{vst} which are needed to ensure the presence of the servicers at the customer satellites where they are assigned to by the optimizer. The idea behind these variables is that setting B_{vst} to 1 should enforce the presence of servicer v at time step t at whatever Customer Node that triggers the need s. Unlike the service assignment variables, these variables are defined for the entire set of time steps $\mathcal{T}$.

Finally, let's define the set $\mathcal{S}_i$ of service needs occurring at customer satellite $i \in \mathcal{C}$, and the set of all service needs $\mathcal{S} = \bigcup_{i \in \mathcal{C}} \mathcal{S}_i$. These sets are indexed by the letter s.

5.2.3.2. OOS infrastructure operation assumptions

Before moving on to the mathematical formulation, the assumptions are presented based on which the objective function and constraints are defined:

1. The launch vehicles are allowed to fly only between the Earth Nodes and the OOS Parking Nodes. This also means that they cannot be staged along the holdover arcs defined at the OOS Parking Nodes. This is to keep the optimizer from leveraging the launch vehicles as makeshift depots;
2. The orbital depots remain staged at the OOS Parking Nodes;
3. One orbital depot is deployed at every OOS Parking Node defined in the network. This allows to model sustainable OOS infrastructures that support the long-term operations of servicers (e.g., through regular servicer refueling);
4. Each orbital depot is assumed to be a small-scale robotic space station technically capable of operating the transfer of non-vehicle commodities across the different spacecraft (e.g., from the fairing of a launch vehicle to a servicer's cargo bay);
5. The servicers are not capable of exchanging commodities with each other. They may exchange non-vehicle commodities at the OOS Parking Nodes through the orbital depots;
6. There cannot be more than one servicer at a time at a Customer Node. This prevents undesirable traffic of servicers near customer satellites;
7. A servicer can provide only one service at a time to a single customer satellite, even if it is equipped with several tools;
8. A servicer cannot depart the OOS Parking Node at which it is staged unless it has been assigned a service need;
9. A servicer must go back to one of the available OOS Parking Nodes after providing a service unless it has been assigned a sequence of back-to-back services;
10. The operating costs of the servicers are the same no matter whether they are idle at a node, flying between two different nodes, or providing a service to a customer

satellite;

11. No penalty fee is incurred to the OOS infrastructure if the optimizer decides not to address a service need. Penalty fees, however, are incurred if the optimizer decides to provide a service and does so with delays. Here, the delay is defined as the difference between the time step at which a service need is triggered and the time step at which the service starts.

5.2.3.3. Objective function

This chapter assumes that the goal of OOS businesses is to maximize their profits, defined as the revenues generated by the provision of services minus the costs to manufacture, deploy and operate the OOS infrastructures. The revenues in the MILP model are captured by Equation 5.1,

$$J_{\text{revenues}} = \sum_{v\in\mathcal{V}_{\text{serv}}} \sum_{i\in\mathcal{C}} \sum_{s\in\mathcal{S}_i} \sum_{t\in\mathcal{W}_s} \left(r_s H_{vst}\right), \tag{5.1}$$

where r_s is the revenue generated if service need s is addressed by the OOS infrastructure.

Equation 5.2 gives the expression for the launch cost,

$$J_{\text{launch}} = c^{\text{launch}} \sum_{v\in\mathcal{V}} \sum_{t\in\mathcal{T}} \sum_{i\in\mathcal{E}} \sum_{j\in\mathcal{P}} \left(\sum_{k\in\mathcal{K}_{\text{cont}}} U^{+}_{vijq_0r_0tk} + \sum_{k\in\mathcal{K}_{\text{int}}} m_k U^{+}_{vijq_0r_0tk} + m_v W^{+}_{vijq_0r_0t} \right), \tag{5.2}$$

where m_k and m_v are the mass per unit of integer commodity k and vehicle v respectively, c^{launch} is the launch cost per unit mass, and q_0 and r_0 are special indices used to characterize the launches from the Earth Nodes to the OOS Parking Nodes.

Equation 5.3 expresses the cost associated with developing and manufacturing the servicing elements, and purchasing the consumables to support the OOS infrastructure,

$$J_{\text{pdm}} = \sum_{v\in\mathcal{V}} \sum_{t\in\mathcal{T}} \sum_{i\in\mathcal{E}} \sum_{j\in\mathcal{P}} \left(\sum_{k} c_k^{\text{pdm}} U^{+}_{vijq_0r_0tk} + c_v^{\text{pdm}} W^{+}_{vijq_0r_0t} \right), \tag{5.3}$$

where c_k^{pdm} is the cost per unit of integer commodity and per unit mass of continuous commodity, c_v^{pdm} is the cost per unit of vehicle commodity, and the acronym "pdm" stands for *purchase/development/manufacturing* costs.

Equation 5.4 captures the fees incurred to the OOS infrastructure if a service is provided with delays,

$$J_{\text{delay}} = \sum_{i \in \mathcal{C}} \sum_{s \in \mathcal{S}_i} \sum_{\tau \in \mathcal{W}_s} \left(c_s^{\text{delay}} (\tau - \tau_s) \sum_{v \in \mathcal{V}_{\text{serv}}} H_{vs\tau} \right), \tag{5.4}$$

where c_s^{delay} is the fee per unit time of delay, and τ_s is the time step representing the beginning of the service window associated with service need s. Note that τ_s is the smallest of the time steps defined in $\mathcal{W}_s$, so the delay $\tau - \tau_s$ with $\tau_s \in \mathcal{W}_s$ is nonnegative.

Equation 5.5 models the operating cost of the orbital depots,

$$J_{\text{dep}} = \sum_{v \in \mathcal{V}_{\text{dep}}} \sum_{i \in \mathcal{P}} \sum_{t \in \mathcal{T}} \left(c_v^{\text{dep}} \Delta_t Y_{vit}^{+} \right), \tag{5.5}$$

where $c_v^{\text{dep}}$is the operating cost of the depots per unit time, and Δ_t is the length of the holdover arc starting at time step t.

Finally, Equation 5.6 models the operating cost of the servicers,

$$J_{\text{serv}} = \sum_{v \in \mathcal{V}_{\text{serv}}} \sum_{i \in \mathcal{N}_{\text{orb}}} \sum_{t \in \mathcal{T}} c_v^{\text{serv}} \left(\Delta_t Y_{vit}^{+} + \sum_{\substack{j \in \mathcal{N}_{\text{orb}} \\ j \neq i}} \sum_{q \in \mathcal{Q}_v} \sum_{r \in \mathcal{R}_{vq}} q W_{vijqrt}^{+} \right), \tag{5.6}$$

where c_v^{serv} is the operating cost of the servicers per unit time.

5.2.3.4. Mass balance constraints

The mass balance constraints ensure the conservation of commodities at the nodes of the network. Equation 5.7 first gives the mass balance of non-vehicle commodities at the Customer Nodes. This constraint is written per servicer to model the fact that servicers cannot exchange commodities with each other. Note also that the mass balance constraint at the

Customer Nodes is not specialized for launch vehicles and orbital depots as these spacecraft are not allowed to visit the customer satellites.

$$\forall v \in \mathcal{V}_{\text{serv}},\ \forall i \in \mathcal{C},\ \forall t \in \mathcal{T}:$$
$$\boldsymbol{X}^{+}_{vit} - \boldsymbol{X}^{-}_{vi(t-\Delta'_t)} + \sum_{\substack{j\in\mathcal{N}\\ q\in\mathcal{Q}_v\\ r\in\mathcal{R}_{vq}\\ i\neq j}} \left(\boldsymbol{U}^{+}_{vijqrt} - \boldsymbol{U}^{-}_{vjiqr(t-q)}\right) = \begin{cases} \sum_{s\in\mathcal{S}_i} \boldsymbol{d}_{vs} H_{vst} & \text{if } t \in \mathcal{L}_i \\ 0 & \text{otherwise} \end{cases}, \tag{5.7}$$

where Δ'_t is the length of the holdover arc directly preceding time step t, $\boldsymbol{d}_{vs}$ is the column vector of nonpositive demand for the non-vehicle commodities (e.g., propellant for refueling), and $\mathcal{L}_i$ is the set of time steps which fall within at least one of the service windows $\mathcal{W}_s$ defined for the service needs s occurring at customer satellite i. The set $\mathcal{L}_i$ is defined in Equation 5.8:

$$\mathcal{L}_i = \bigcup_{s\in\mathcal{S}_i} \mathcal{W}_s. \tag{5.8}$$

Equation 5.9 gives the mass balance of the non-vehicle commodities at the OOS Parking Nodes. Note that unlike Equation 5.7, this equation is written by summing the variables over the vehicle index $v \in \mathcal{V}$ to allow the spacecraft to exchange commodities with each other through the orbital depots,

$$\forall i \in \mathcal{P},\ \forall t \in \mathcal{T}:$$
$$\sum_{v\in\mathcal{V}} \left(\boldsymbol{X}^{+}_{vit} - \boldsymbol{X}^{-}_{vi(t-\Delta'_t)}\right) + \sum_{\substack{v\in\mathcal{V}\\ j\in\mathcal{N}\\ q\in\mathcal{Q}_v\\ r\in\mathcal{R}_{vq}\\ i\neq j}} \left(\boldsymbol{U}^{+}_{vijqrt} - \boldsymbol{U}^{-}_{vjiqr(t-q)}\right) = 0. \tag{5.9}$$

Equation 5.10 models the supply of commodities from the Earth Nodes, with $\boldsymbol{\sigma}_{it}$ being the column vector that represents the nonnegative supply of non-vehicle commodities,

$$\forall i \in \mathcal{E},\ \forall t \in \mathcal{T}:$$
$$\sum_{v\in\mathcal{V}} \left(\boldsymbol{X}^{+}_{vit} - \boldsymbol{X}^{-}_{vi(t-\Delta'_t)}\right) + \sum_{\substack{v\in\mathcal{V}\\ j\in\mathcal{N}\\ q\in\mathcal{Q}_v\\ r\in\mathcal{R}_{vq}\\ i\neq j}} \left(\boldsymbol{U}^{+}_{vijqrt} - \boldsymbol{U}^{-}_{vjiqr(t-q)}\right) < \boldsymbol{\sigma}_{it}. \quad (5.10)$$

Finally, Equation 5.11 and Equation 5.12 represent the mass balance of the non-vehicle commodity for each type of spacecraft at the Orbital Nodes (i.e., Customer Nodes and OOS Parking Nodes) and Earth Nodes, respectively,

$$\forall v \in \mathcal{V},\ \forall i \in \mathcal{N}_{\text{orb}},\ \forall t \in \mathcal{T}:$$
$$\left(Y^{+}_{vit} - Y^{-}_{vi(t-\Delta'_t)}\right) + \sum_{\substack{j\in\mathcal{N}\\ q\in\mathcal{Q}_v\\ r\in\mathcal{R}_{vq}\\ i\neq j}} \left(W^{+}_{vijqrt} - W^{-}_{vjiqr(t-q)}\right) = 0, \quad (5.11)$$

$$\forall v \in \mathcal{V},\ \forall i \in \mathcal{E},\ \forall t \in \mathcal{T}:$$
$$\left(Y^{+}_{vit} - Y^{-}_{vi(t-\Delta'_t)}\right) + \sum_{\substack{j\in\mathcal{N}\\ q\in\mathcal{Q}_v\\ r\in\mathcal{R}_{vq}\\ i\neq j}} \left(W^{+}_{vijqrt} - W^{-}_{vjiqr(t-q)}\right) \leq \sigma'_{ivt}, \quad (5.12)$$

where σ'_{ivt} represents the nonnegative supply of spacecraft at the Earth Nodes.

5.2.3.5. Concurrency constraints

The concurrency constraints relate to the payload capacities of the vehicles considered in the network. This is to enforce the transportation of commodities (e.g., propellant, spares) within vehicles. Equation 5.13 represents the concurrency constraints along holdover arcs, while Equation 5.14 is for the concurrency constraints along transportation arcs.

$$\forall v \in \mathcal{V},\ \forall i \in \mathcal{N}_{\text{orb}},\ \forall t \in \mathcal{T}: \qquad \overline{\overline{M}}_{vii} \boldsymbol{X}^{+}_{vit} \leq \boldsymbol{e}_v Y^{+}_{vit}, \tag{5.13}$$

$$\forall v \in \mathcal{V},\ \forall i \in \mathcal{N},\ \forall j \in \mathcal{N},\ i \neq j,\ \forall q \in \mathcal{Q}_v,\ \forall r \in \mathcal{R}_{vq},\ \forall t \in \mathcal{T}:$$
$$\overline{\overline{M}}_{vij} \boldsymbol{U}^{+}_{vijqrt} \leq \boldsymbol{e}_v W^{+}_{vijqrt}, \tag{5.14}$$

where $\overline{\overline{M}}_{vij}$ is the concurrency matrix, and $\boldsymbol{e}_v$ represents the vehicle design parameters such as payload and propellant capacities.

5.2.3.6. *Commodity transformation constraints*

These constraints model the consumption of the commodities needed to operate the depots and servicers (e.g., propellant consumption). Equation 5.15 and Equation 5.16 give the constraints for holdover and transportation arcs respectively when the consumption models are linear:

$$\forall v \in \mathcal{V},\ \forall i \in \mathcal{N},\ \forall t \in \mathcal{T}: \qquad \overline{\overline{Q}}'_{vi} \begin{bmatrix} \boldsymbol{X}^{+}_{vit} \\ Y^{+}_{vit} \end{bmatrix} = \begin{bmatrix} \boldsymbol{X}^{-}_{vit} \\ Y^{-}_{vit} \end{bmatrix}, \tag{5.15}$$

$$\forall v \in \mathcal{V},\ \forall i \in \mathcal{N},\ \forall j \in \mathcal{N},\ i \neq j,\ \forall q \in \mathcal{Q}_v,\ \forall r \in \mathcal{R}_{vq},\ \forall t \in \mathcal{T}:$$
$$\overline{\overline{Q}}''_{vijqr} \begin{bmatrix} \boldsymbol{U}^{+}_{vijqrt} \\ W^{+}_{vijqrt} \end{bmatrix} = \begin{bmatrix} \boldsymbol{U}^{-}_{vijqrt} \\ W^{-}_{vijqrt} \end{bmatrix}, \tag{5.16}$$

where $\overline{\overline{Q}}'_{vi}$ and $\overline{\overline{Q}}''_{vijqr}$ are the mass transformation matrices for holdover and transportation arcs, respectively.

When the consumption models are non-linear, as is often the case of low-thrust models, Equation 5.15 and Equation 5.16 cannot properly capture their non-linear nature. Instead, they must be converted into piecewise linear approximations, and as many additional variables as the number of breakpoints must be introduced. More specifically in this chapter, the model describing the propellant consumption of a servicer with respect to its total initial mass is approximated and integrated into the MILP. $P^{\pm}_{vijqrt}$ is the inflow/outflow variables of the propellant meant to be consumed by the servicers along the transportation arcs. As depicted in Figure 5.4, $F(.)$ denotes the considered non-linear propellant consumption model, and $(b_1, F(b_1)), \ldots, (b_N, F(b_N))$ denote the N breakpoints used to define the piecewise linear approximation. An SOS2[1] set of N continuous nonnegative variables $\lambda_1, \ldots, \lambda_N$ is then introduced, and the approximated value of the propellant consumed over the transportation arc (i.e., $P^{+}_{vijqrt} - P^{-}_{vijqrt}$) can be found as [44]:

$$\forall v \in \mathcal{V}_{\text{serv}},\ \forall i \in \mathcal{N},\ \forall j \in \mathcal{N},\ i \neq j,\ \forall q \in \mathcal{Q}_v,\ \forall r \in \mathcal{R}_{vq},\ \forall t \in \mathcal{T}:$$

$$\lambda_1 + \cdots + \lambda_N = 1, \tag{5.17}$$

$$\lambda_1 b_1 + \cdots + \lambda_N b_N = Z^{+}_{vijqrt}, \tag{5.18}$$

$$\lambda_1 F(b_1) + \cdots + \lambda_N F(b_N) = P^{+}_{vijqrt} - P^{-}_{vijqrt}, \tag{5.19}$$

where Z^{+}_{vijqrt} is the total mass of the servicer along the transportation arc as defined in Equation 5.20. Note that this model is not needed when high-thrust maneuvers are the only trajectory options as the rocket equation yields a linear relationship between the mass of the servicer before an impulse and the propellant consumption to perform that impulse.

$$\forall v \in \mathcal{V}_{\text{serv}},\ \forall i \in \mathcal{N},\ \forall j \in \mathcal{N},\ i \neq j,\ \forall q \in \mathcal{Q}_v,\ \forall r \in \mathcal{R}_{vq},\ \forall t \in \mathcal{T}:$$

$$Z^{+}_{vijqrt} = \sum_{k \in \mathcal{K}_{\text{cont}}} U^{+}_{vijqrtk} + m_v W^{+}_{vijqrtk}. \tag{5.20}$$

[1] An SOS2 (special ordered sets of type 2) constraint enforces that, at most, two of the λ can be nonzero; and these two nonzero elements of the set must be consecutive.

5.2.3.7. Service management constraints

Five constraints are defined for the optimizer to properly manage the services and the associated operations of the fleet of servicers. Equation 5.21 models the assignment of the service needs to the servicers. It ensures that each service need s is assigned at most once to a servicer within the service window $\mathcal{W}_s$.

$$\forall s \in \mathcal{S} : \qquad \sum_{v \in \mathcal{V}_{\text{serv}}} \sum_{\tau \in \mathcal{W}_s} H_{vs\tau} \leq 1. \tag{5.21}$$

Equation 5.22 describes the coupling between the service assignment variables and servicer dispatch variables through the binary parameters $\beta_{s\tau t}$. These parameters are automatically generated by the OOS optimization framework before running the optimization based on the input data related to the service needs. Note that these parameters are indexed by both $\tau \in \mathcal{T}$ and $\tau \in \mathcal{W}_s$. Essentially, the binary parameter $\beta_{s\tau t}$ captures the duration of service s and at which time steps t the service has to be provided if the OOS operator decides to start the service at the date τ.

$$\forall v \in \mathcal{V}_{\text{serv}},\ \forall s \in \mathcal{S},\ \forall t \in \mathcal{T} : \qquad B_{vst} = \sum_{\tau \in \mathcal{W}_s} H_{vs\tau} \beta_{s\tau t}. \tag{5.22}$$

Equation 5.23 is designed to keep the optimizer from providing more than one service at a time to a customer satellite. One and only one service need can be addressed at a time at each Customer Node of the network. This constraint along with Equation 5.24 also avoids the unnecessary clustering of servicers at a Customer Node for the purpose of increased operational safety. Indeed, GEO satellite operators are likely to require minimal servicer traffic around their spacecraft.

$$\forall i \in \mathcal{C},\ \forall t \in \mathcal{T} : \qquad \sum_{v \in \mathcal{V}_{\text{serv}}} \sum_{s \in \mathcal{S}_i} B_{vst} \leq 1. \tag{5.23}$$

$$\forall v \in \mathcal{V}_{\text{serv}}\ \forall i \in \mathcal{C},\ \forall t \in \mathcal{T}: \qquad Y^{+}_{vit} = \sum_{s \in \mathcal{S}_i} B_{vst}. \tag{5.24}$$

Finally, Equation 5.25 ensures the adequate servicer tool is used to provide a service, as specified by the service-tool mapping given in Table 3.5. The binary parameter γ_{sk} is set to 1 if the tool k is required to address service need s, and to 0 otherwise.

$$\forall v \in \mathcal{V}_{\text{serv}}\ \forall i \in \mathcal{C},\ \forall t \in \mathcal{T},\ \forall k \in \mathcal{K}_{\text{tools}}: \qquad X^{+}_{vitk} \geq \sum_{s \in \mathcal{S}_i} \gamma_{sk} B_{vst}. \tag{5.25}$$

5.2.3.8. Servicer spaceflights

Two important constraints are presented here to accurately model the flights of the servicers in a MILP. The first constraint, captured by Equation 5.26, ensures that a servicer cannot fly along a transportation arc between any two nodes of the network if its total mass (i.e., combined servicer's dry mass and payload mass) exceeds the servicer mass upper bound calculated by the corresponding trajectory plugin.

$$\forall v \in \mathcal{V}_{\text{serv}},\ \forall i \in \mathcal{N},\ \forall j \in \mathcal{N},\ i \neq j,\ \forall q \in \mathcal{Q}_v,\ \forall r \in \mathcal{R}_{vq},\ \forall t \in \mathcal{T}:$$
$$Z^{+}_{vijqrt} \leq M^{ub}_{vijqr} W^{+}_{vijqrt}, \tag{5.26}$$

where M^{ub}_{vijqr} is the servicer mass upper bound, and Z^{+}_{vijqrt} is the total mass of the servicer and its payloads as defined by Equation 5.20.

The second constraint, captured by Equation 5.27, makes sure that a servicer departs its parking location only when it is assigned a service need by the optimizer. This is essential for the servicer to go straight from its parking location to its target location without undesirably stopping by other network nodes along the way.

$$\forall v \in \mathcal{V}_{\text{serv}},\ \forall i \in \mathcal{C},\ \forall t \in \mathcal{T}:$$

$$\sum_{\substack{j \in \mathcal{N} \\ q \in \mathcal{Q}_v \\ r \in \mathcal{R}_{vq} \\ j \neq i}} W^{+}_{vjiqr(t-q)} = \begin{cases} \sum_{s \in \mathcal{S}_i} H_{vst} & \text{if } t \in \mathcal{L}_i \\ 0 & \text{otherwise} \end{cases}. \tag{5.27}$$

5.3 Trajectory Models

The formulation proposed in Section 5.2 enables the trajectory models to be used for the OOS mission design optimization. Although the proposed formulation can integrate various trajectory plugins, this section describes examples of high-thrust and low-thrust models that can be integrated into the trajectory plugins of the OOS logistics framework. The procedure needed to derive the propellant consumption as a function of the servicer's initial mass is given for both models. Note that through the standard interface between the OOS logistics framework and the trajectory plugins, the users of the framework can define as many new trajectory models as desired.

5.3.1 High-thrust phasing maneuver

The high-thrust phasing maneuver considered in this chapter is a two-impulse rendezvous maneuver. The delta V of the trajectory is computed as described in subsection 4.1.2. The rocket equation relates the consumed propellant m_p as a function of the initial mass m_0 of the servicer (i.e., dry mass and cargo) and delta-V of the maneuver. This is given in Equation 5.28.

$$m_p = m_0 \left(1 - \exp\left(-\frac{\Delta V}{g_0 I_{sp}}\right)\right) \tag{5.28}$$

where g_0 is the gravitational acceleration of the Earth at sea level, and I_{sp} is the specific impulse of the high-thrust engine. Note that this model is linear and thus does not require the plugin to find its piecewise linear approximation. Since this model is valid for any initial mass m_0 of the servicer, the trajectory plugin would output the maximum initial mass of the servicer as the servicer mass upper bound (i.e., using the notations in Figure 5.4, $M^{ub} = M_{max}$).

5.3.2 Low-thrust phsaing maneuver

The analytical low-thrust trajectory model presented in this subsection is based on Ref. [45] (cf pp. 5-7, Re-positioning in Orbits: Walking). This trajectory is a phasing maneuver for a low-thrust servicer to change its angular position along a circular orbit (e.g. GEO orbit). It consists of three phases: (1) an initial thrust phase which brings the servicer from its initial circular orbit to an intermediate circular orbit; (2) a coasting phase along the intermediate circular orbit; and (3) a final thrust phase that brings the servicer back to its initial circular orbit but at a different angular position. Figure 5.5 illustrates the trajectory. Note that due to the low thrust of the servicer, the initial and final thrust phases are spiraling about the Earth but are not represented as such in Figure 5.5 for clarity. The phasing angle $\delta\theta$ defined in Figure 5.5 is the angle between the position the servicer would have if it had remained coasting along its initial circular orbit and its actual position with the phasing maneuver. The phasing angle is represented at three different dates t_1, t_2, t_3 in Figure 5.5.

A few assumptions are made in Ref. [45] to develop a simple analytical model describing the low-thrust phasing maneuver introduced in Figure 5.5:

- The thrust is constant, continuous, and tangential during the thrusting phases;
- The mass of the servicer is assumed constant during the maneuver due to a low propellant consumption. This results in a constant thrust acceleration during the thrusting phases;
- The thrust acceleration is assumed much smaller than the gravitational and centripetal

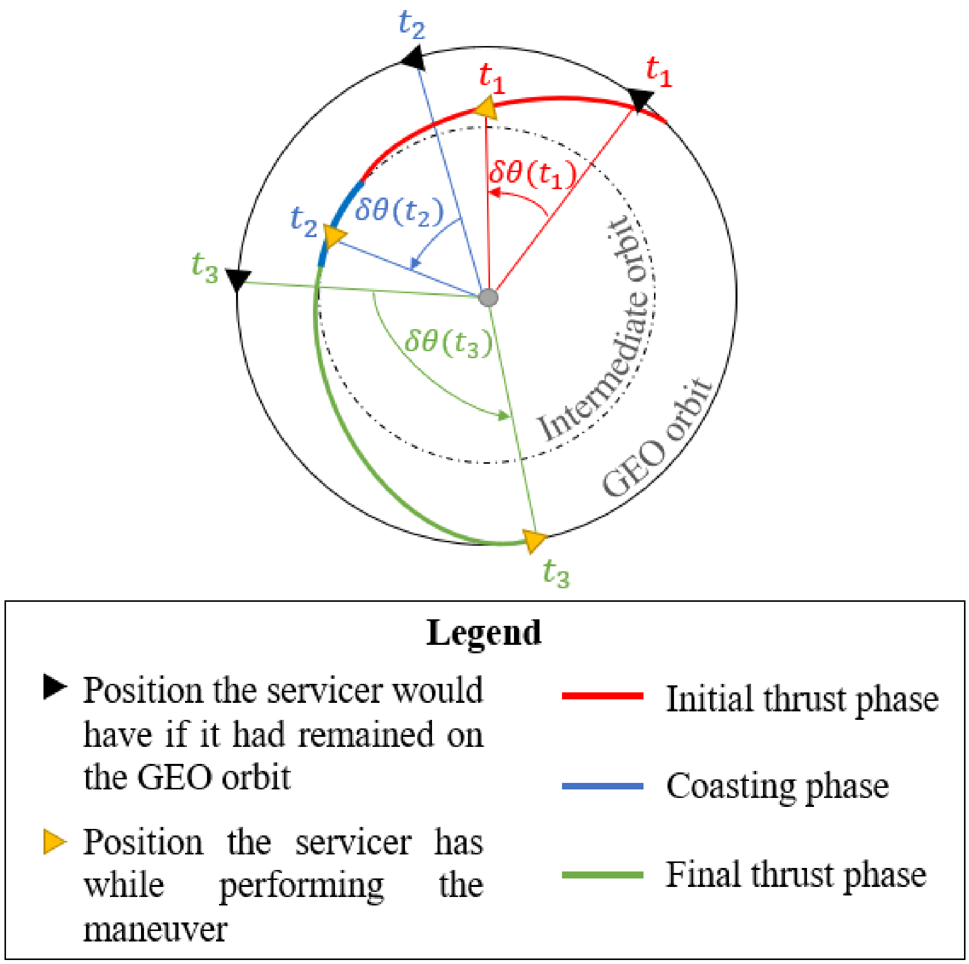

Figure 5.5: Overview of the low-thrust phasing maneuver modeled in this chapter.

accelerations to approximate the spiraling transfer as a series of circular orbits of varying radii (spiral approximation);

- The specific impulse remains constant during the thrusting phases (i.e., the engine runs at a constant power level).

From these assumptions and the power balance, Ref. [45] develops Equation 5.29 describing the dynamic evolution of the phasing angle,

$$\frac{d^2(\delta\theta)}{dt^2} = -\frac{3a_\theta}{r_0} \tag{5.29}$$

where a_θ is the thrust acceleration due to the engine, and r_0 is the radius of the initial circular orbit. Note that the thrust acceleration is backward (i.e., negative) for a positive target phasing angle, and forward (i.e., positive) otherwise. If $\Delta\theta$ is the target phasing angle defined over $(-\pi, \pi]$, then the thrust acceleration is given by Equation 5.30 as

$$a_\theta = \text{sign}(\Delta\theta)\frac{F}{m_0} \tag{5.30}$$

Equation 5.30 is then integrated twice for each phase of the maneuver and the continuity in the servicer's phasing angle and rate between the different phases is enforced. This leads to Equation 5.31, which is a quadratic equation in the duration τ of the initial and final thrust phases. Note that since the mass of the servicer is assumed constant during the maneuver, the initial and final thrust phases are of the same duration τ.

$$\tau^2 - t_f\tau + \frac{r_0 m_0|\Delta\theta|}{3F} = 0 \tag{5.31}$$

where t_f is the duration of the entire rendezvous maneuver.

The coefficients of Equation 5.31 are known: $\Delta\theta$, t_f, r_0 and F are the inputs to the trajectory plugin, and m_0 is the initial mass of the servicer, which is varied between some bounds M_{min} and M_{max} by the trajectory plugin to compute the propellant consumption model associated with the low-thrust phasing maneuver. The servicer mass upper bound M^{ub} is then found by setting the discriminant of Equation 5.31 to 0 and solving for the initial mass. The flight of the servicer is feasible for an initial mass $m_0 \leq M^{ub}$ but infeasible otherwise as modeled by Equation 5.26.

Assuming that the maneuver is feasible for the given initial mass m_0, the trajectory plugin proceeds by computing the propellant consumed during the maneuver assuming a constant propellant mass flow rate $b > 0$,

$$m_p = 2b\tau \tag{5.32}$$

From the thrust force F and specific impulse I_{sp} of the low-thrust engine (also an input to the trajectory plugin), the expression for the propellant mass flow rate is:

$$b = \frac{F}{g_0 I_{sp}} \tag{5.33}$$

The non-linear propellant consumption model is computed and illustrated in Figure 5.6 for a constant thrust force F=1.16N, a specific impulse I_{sp}=1,790s, a time of flight Δt=8 days, a target phasing angle $\Delta\theta$=180 deg, and an initial orbit radius r_0=42,164 km corresponding to the GEO orbit. The data for the thrust force and specific impulse are based upon Northrop Grumman's MEV [10] [46]. The model is computed within the bounds M_{min} = 500 kg and M_{max} = 4,000 kg for the servicer's initial mass. The servicer mass upper bound is found to be 3,138 kg as can be seen at the discontinuity in Figure 5.6. For a servicer initial mass below 3,138 kg, the maneuver is feasible, but infeasible beyond. In addition to computing this non-linear model, the trajectory plugin would proceed by finding its piecewise linear approximation for proper integration into the MILP.

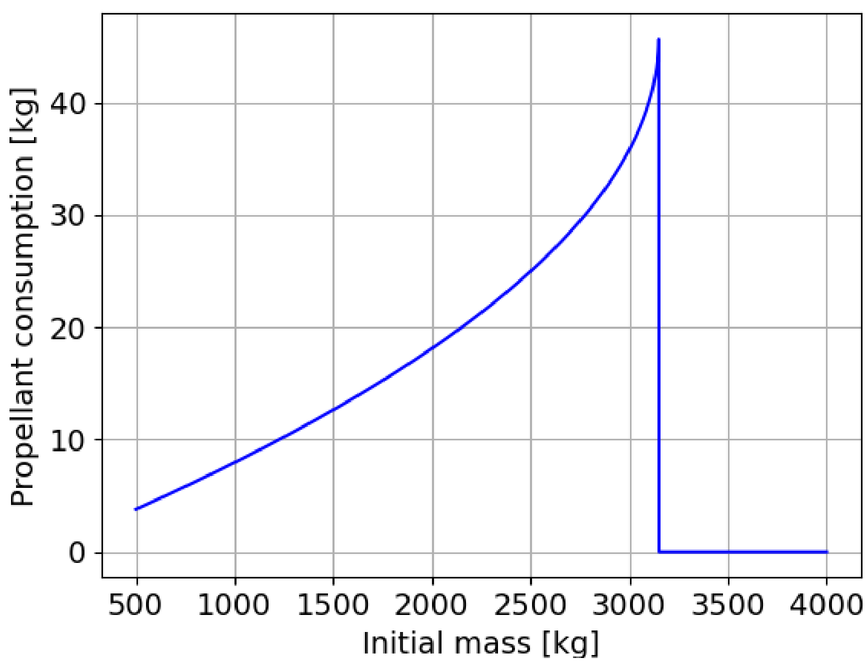

Figure 5.6: Non-linear propellant consumption of the servicer as a function of its initial mass.

5.4 Application

This section demonstrates two use cases of the OOS logistics optimization framework: a short-term operational scheduling use case; and a long-term strategic planning use case. In both cases, the optimizer now has the capability to automatically trade between the high- and low-thrust propulsion systems and trajectories of the servicers. Subsection 5.4.1 summarizes the assumptions used to run the different scenarios designed to demonstrate the use cases. Subsection 5.4.2 demonstrates the proposed method as a tool for short-term operational scheduling of an existing OOS infrastructure equipped with both high- and low-thrust technologies. Subsection 5.4.3 demonstrates the proposed method as a tool to inform long-term strategic design of OOS architectures, especially regarding high- and low-thrust technologies, based on their performance given various OOS market conditions.

5.4.1 Assumptions and scenarios

In this subsection, the assumptions and data associated with the fleet of customer satellites are presented. Then, the assumptions and data associated with the OOS infrastructure are given. Finally, the scenarios associated with the two use-cases are presented.

Customer fleet assumptions

The data related to the deterministic and random service needs are given in Figure 4.14 and Figure 4.15, respectively; these are the same assumptions used in the case studies of chapter 4 but are repeated here for convenience. Note that the given data are for one customer satellite; by increasing the number of customer satellites in the simulations, the service need rates of the entire fleet of customer satellites increase. Note that in the tables, data without reference is assumed. Finally, the positions of the customer satellites were retrieved from the UCS (Union of Concerned Scientists) database of satellites [40].

Table 5.2: Assumptions related to the deterministic service needs.

	Inspection	Refueling	Station Keeping
Revenues	10 [17]	15 [17]	20
Delay penalty fee [$/day]	5,000 [17]	100,000 [17]	100,000 [17]
Service duration [days]	10 [17]	30 [17]	180 [24]
Service window [days]	30	30	30
Frequency of occurrence [days]	6,310 [21]	2,100 [21]	2,100 [21]

Table 5.3: Assumptions related to the random service needs.

	Repositioning	Retirement	Repair	Mechanism Deployment
Revenues	10 [17]	10 v	30	25 [17]
Delay penalty fee [$/day]	100,000 [17]	0 [17]	100,000 [17]	100,000 [17]
Service duration [days]	30 [17]	30 [17]	30 [17]	30 [17]
Service window [days]	30	30	30	30
Mean frequency of occurrence [days]	2,520 [21]	2,520 [21]	9,020 [21]	21,050 [21]

OOS infrastructure assumptions

The four notional servicer tools given in Table 3.5 have an assumed cost of $100,000 and an assumed mass of 100kg. The other commodities considered in the case studies are the spares (assumed price tag: $1,000/kg), bipropellant for the servicers (price tag for Monomethyl Hydrazine: $180/kg), monopropellant (price tag for Hydrazine: $230/kg), and low-thrust propellant (price tag for Xenon gas: $1,115/kg).

An orbital depot is assumed pre-deployed at a GEO orbital slot located at a longitude of 170 deg West (over the Pacific Ocean). The depot is assumed to consume its own monopropellant at a rate of 0.14kg/day for station keeping [43]. The manufacturing and operating costs of the depot are assumed to be $200M and $13,000/day, respectively.

The launch vehicle used for this analysis is based on a Falcon 9 launcher with an assumed maximum payload capacity of 8,300kg. A launcher is assumed to be available every 30 days for resupply of the depot. The mass-specific launch cost is assumed to be $11,300/kg.

Six different servicer designs are simulated in this chapter based on their propulsion systems and the number of tools they are integrated with. Three types of servicers are defined based on the propulsion system they use to fly: high-thrust servicers; low-thrust servicers; and multimodal servicers, which integrate both high- and low-thrust propulsion technologies. Two subtypes are then defined based on the number of tools they are equipped with: one versatile servicer, and four specialized servicers. The versatile servicer can provide all seven defined services whereas the specialized servicers can only provide the services for which their tools are suited. The detailed assumptions are given in Table 5.4, Table 5.5, and Table 5.6 for the high-thrust servicers, low-thrust servicers, and multimodal servicers, respectively. The baseline dry mass for the high-thrust versatile, low-thrust versatile, and multimodal specialized servicers is taken from [17]. This baseline mass is decreased for the high-thrust and low-thrust specialized servicers which are assumed to be less capable and smaller in size than their versatile counterparts. Similarly, this baseline mass is increased for the multimodal versatile servicers because they integrate an additional propulsion system compared to their high- and low-thrust versatile counterparts. The model for the low-thrust propulsion system is based on Northrop Grumman's MEV system of four XR-5 Hall Thrusters [10] [46].

Finally, this chapter assumes the refueling of the servicers to be instantaneous operations. The justification behind this assumption is that, as OOS operations become routine, refueling of the servicers will likely not take more than one time step in the dynamic network (i.e., two days) [41]. This assumption can be modified depending on the technology performance.

Case studies' scenarios

Using the assumptions presented previously, several scenarios are designed to demonstrate the framework's value in supporting trade studies related to the propulsion technologies of the servicers. Two case studies are considered in this chapter: the short-term operational scheduling of a multimodal servicer; and the long-term strategic planning of six different OOS architectures. Three different market conditions are defined by considering 30 satel-

Table 5.4: Assumptions related to the high-thrust servicers

	Versatile	Specialized 1	Specialized 2	Specialized 3	Specialized 4
Tools	T1, T2, T3, T4	T1	T2	T3	T4
Dry mass [kg]	3,000 [17]	2,000	2,000	2,000	2,000
Propellant capacity [kg]	1,000	1,000	1,000	1,000	1,000
Manufacturing cost [$M]	75	50 [42]	50 [42]	50 [42]	50 [42]
Operating cost [$/day]	13,000 [25]	13,000 [25]	13,000 [25]	13,000 [25]	13,000 [25]
Propellant type	Bi-propellant	Bi-propellant	Bi-propellant	Bi-propellant	Bi-propellant
Specific Impulse [s]	316	316	316	316	316
Flight durations [days]	2, 4	2, 4	2, 4	2, 4	2, 4

Table 5.5: Assumptions related to the low-thrust servicers

	Versatile	Specialized 1	Specialized 2	Specialized 3	Specialized 4
Tools	T1, T2, T3, T4	T1	T2	T3	T4
Dry mass [kg]	3,000 [17]	2,000	2,000	2,000	2,000
Propellant capacity [kg]	300	300	300	300	300
Manufacturing cost [$M]	75	50 [42]	50 [42]	50 [42]	50 [42]
Operating cost [$/day]	13,000 [25]	13,000 [25]	13,000 [25]	13,000 [25]	13,000 [25]
Propellant type	Low-thrust propellant	Low-thrust propellant	Low-thrust propellant	Low-thrust propellant	Low-thrust propellant
Specific Impulse [s]	1,790 [10] [46]	1,790 [10] [46]	1,790 [10] [46]	1,790 [10] [46]	1,790 [10] [46]
Thrust [N]	1.74 [10] [46]	1.74 [10] [46]	1.74 [10] [46]	1.74 [10] [46]	1.74 [10] [46]
Flight durations [days]	10, 14, 30, 34	10, 14, 30, 34	10, 14, 30, 34	10, 14, 30, 34	10, 14, 30, 34

Table 5.6: Assumptions related to the multimodal servicers

	Versatile	Specialized 1	Specialized 2	Specialized 3	Specialized 4
Tools	T1, T2, T3, T4	T1	T2	T3	T4
Dry mass [kg]	4,000	3,000	3,000	3,000	3,000
Bi-propellant capacity [kg]	1,000	1,000	1,000	1,000	1,000
Low-thrust propellant capacity [kg]	300	300	300	300	300
Manufacturing cost [$M]	100	75	75	75	75
Operating cost [$/day]	13,000	13,000	13,000	13,000	13,000
HIGH-THRUST PARAMETERS					
Propellant type	Bi-propellant	Bi-propellant	Bi-propellant	Bi-propellant	Bi-propellant
Specific Impulse [s]	316	316	316	316	316
Flight durations [days]	2, 4	2, 4	2, 4	2, 4	2, 4
LOW-THRUST PARAMETERS					
Propellant type	Low-thrust propellant	Low-thrust propellant	Low-thrust propellant	Low-thrust propellant	Low-thrust propellant
Specific Impulse [s]	1,790 [10] [46]	1,790 [10] [46]	1,790 [10] [46]	1,790 [10] [46]	1,790 [10] [46]
Thrust [N]	1.74 [10] [46]	1.74 [10] [46]	1.74 [10] [46]	1.74 [10] [46]	1.74 [10] [46]
Flight durations [days]	10, 14, 30, 34	10, 14, 30, 34	10, 14, 30, 34	10, 14, 30, 34	10, 14, 30, 34

lites (low demand rate), 71 satellites (medium demand rate), or 142 satellites (high demand rate).

The first case study aims to demonstrate the optimizer's ability to automatically trade between the high- and low-thrust engines of a multimodal servicer to maximize the short-term profits of an OOS venture. The regular scheduling of the short-term operations of OOS infrastructures will be essential to account for random demand (e.g., repair need) and remain competitive. In this first case study, the servicer is assumed to be initially pre-deployed at some customer satellite to account for the fact that it may already be providing services when the re-planning event occurs. The framework is run for a single planning horizon of the RH procedure. This scenario is summarized in Table 5.7.

The second case study aims to compare the long-term value (i.e., profits minus initial investments) of six different OOS architectures leveraging various propulsion technologies. This is done by running the OOS logistics framework for a five-year time horizon, while leveraging the RH procedure embedded in the framework to account for the random service

Table 5.7: Scenarios definition for the short-term operational scheduling case study.

Servicers	1 multimodal versatile servicer
Depot	1 depot pre-deployed at 170 deg West on the GEO orbit
Planning horizon	90 days
Customer fleet	142 customer satellites (high demand)
Initial conditions	The servicer is initially deployed at the SBIR GEO 2 satellite due to a prior service

Table 5.8: Scenarios definition for the long-term strategic planning case study.

	Arch. 1 (high-thrust monolithic)	**Arch. 2 (high-thrust distributed)**	**Arch. 3 (low-thrust monolithic)**	**Arch. 4 (low-thrust distributed)**	**Arch. 5 (multimodal monolithic)**	**Arch. 6 (multimodal distributed)**
Number of servicers	1	4	1	4	1	4
Servicer propulsion	High-thrust	High-thrust	Low-thrust	Low-thrust	Multimodal	Multimodal
Servicer platform design	Versatile (4 tools per servicer)	Specialized (1 tool per servicer)	Versatile (4 tools per servicer)	Specialized (1 tool per servicer)	Versatile (4 tools per servier)	Specialized (1 tool per servicer)

needs. For this case study, the value of six different OOS architectures under low service demand (30 satellites), medium service demand (71 satellites), and high service demand (142 satellites) is assessed. The architectures, given in Table 5.8, are defined based on the servicers' propulsion technologies (i.e., high-thrust, low-thrust, or multimodal propulsion) and platform designs (i.e., versatile or specialized). Architectures involving one versatile servicer are referred to as monolithic, while those involving four specialized servicers are referred to as distributed. As in the first case study, an orbital depot is pre-deployed at a longitude of 170deg West on the GEO orbit. Finally, note that, although this is not demonstrated in this chapter, the proposed framework can simulate OOS architectures involving servicers with different propulsion technologies (e.g., low-thrust and high-thrust servicers working in concert to provide services).

5.4.2 Case study 1: short-term operational scheduling

Whenever an OOS operator decides to re-plan the operations of their infrastructure, they would run the proposed framework with an updated set of service needs and the initial conditions corresponding to the state of the infrastructure at re-planning. In the output, they would then get a breakdown of the servicers' operations over the planning horizon to inform short-term decision making.

The present case study is run for a customer base of 142 customer satellites. However, not all of these satellites will need a service during the period considered for re-planning. The framework thus automatically includes in the network only the satellites that display service needs over the planning horizon. This minimizes the size of the optimization problem thus allowing for a shorter computational time. For example, in the present case study, out of the 142 customer satellites, only 19 exhibit at least one service need, five of which are actually visited by the multimodal versatile servicer in the optimal solution. Figure 5.7 gives the network for the considered scenario. The customer satellites are represented by colored dots. The red dots correspond to the visited customer satellites, whereas the blue dots are not visited by the servicer. The black star in Figure 5.7 represents the orbital depot.

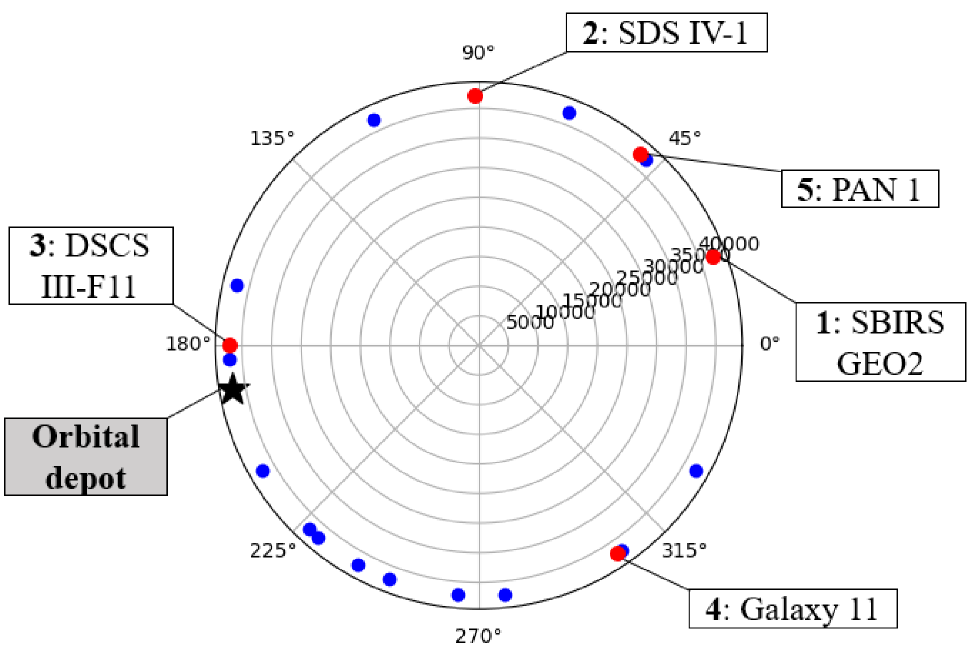

Figure 5.7: Static network used by the optimizer to plan the short-term operations of a multimodal versatile servicer.

The optimization framework is run on an Intel® Core™ i7-9700, 3.00GHz platform with the Gurobi 9 optimizer. The solution was reached in 45 seconds with a gap of 1% a stopping criterion. Figure 5.8, Figure 5.9, and Figure 5.10 illustrate the optimal operations of the multimodal versatile servicer over time.

As seen on those figures, the optimizer has the servicer fly with both high- and low-thrust propulsion systems depending on the time the servicer has left to travel between the nodes of the network. The high- and low-thrust transportation arcs of the servicer are represented with different dashed lines on the figures. Note that at $t = 50$, the servicer leaves the DSCS III-F11 satellite for the depot, and gets refueled in both low-thrust propellant and bi-propellant in order to perform the next low-thrust maneuver at $t = 54$ and high-thrust maneuver at $t = 70$.

For information, the operations presented on the figures lead to a maximized profit of $58.8M over this 90-day period. The servicer was able to capture $75M in revenues from the provision of four services (to, in chronological order, SDS IV-1, DSCS III-F11, Galaxy 11, and PAN 1) for a total cost of $16.2M, including launch, purchase of propellant, delay penalties, and infrastructure operations. Note that these values may not be final for this 90-day period as random service needs may occur within that period, thus triggering future re-planning events and likely modifying the short-term profits.

5.4.3 Case study 2: long-term strategic planning

Careful long-term strategic planning is critical for new entrants in the OOS market to design servicing infrastructures that are resilient to competition and variations in service demand and can generate significant marginal profits in the prevision of market growth. Decision-makers will need to explore a large tradespace capturing both the operational strategies and the designs of the servicers and orbital depots, while considering the uncertainties in service demand. The generalization of the framework to enable the tradeoff between high- and low-thrust technologies is believed to be a significant step forward toward that goal. On the other hand, the RH procedure is critical to properly account for random service needs,

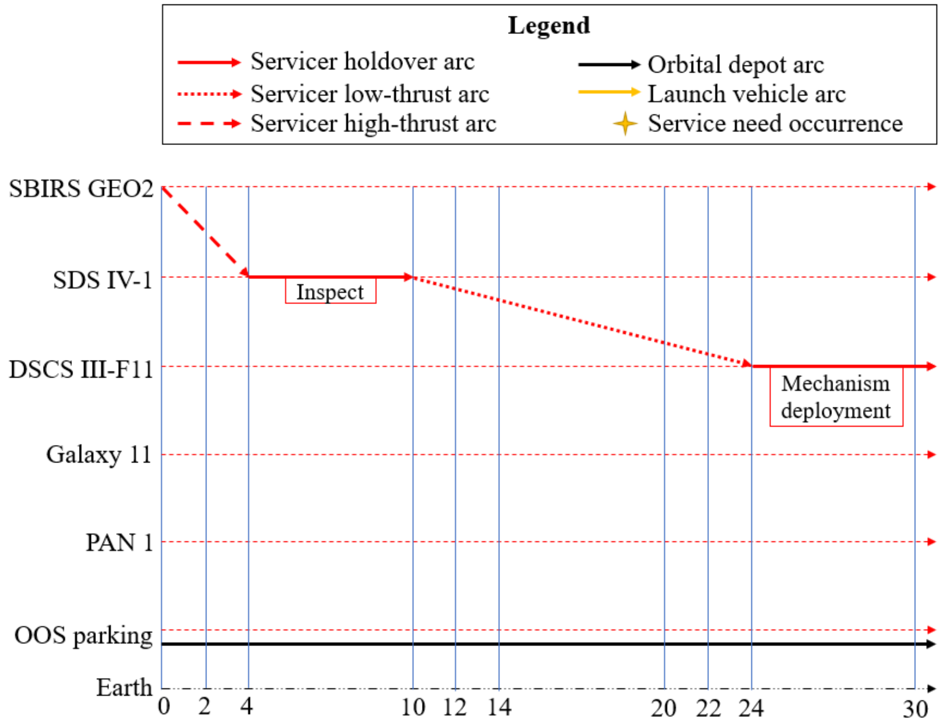

Figure 5.8: Breakdown over the first month (days 0-30) of the operations of an OOS infrastructure consisting of a multimodal versatile servicer and an orbital depot.

which, in essence, are nothing more than unplanned sources of revenues.

For this case study, three different analyses are performed. First, the values of the monolithic and distributed architectures for each propulsion technology and different market conditions (30, 71, 142 customer satellites) are compared. Then, the impact of the servicers' propulsion technologies on the value of the OOS infrastructures is discussed. Finally, the sensitivity of the value of an OOS infrastructure to variations in the mass of the servicers is assessed.

Trading architectural options: monolithic vs distributed architectures

The purpose of this section is to compare the performance of the monolithic and distributed architectures per propulsion system. More specifically, Architecture 1 is compared with Architecture 2, Architecture 3 with Architecture 4, and Architecture 5 with Architecture 6 (cf Table 5.8) for three different levels of service demands (30, 71, 142 customer satellites).

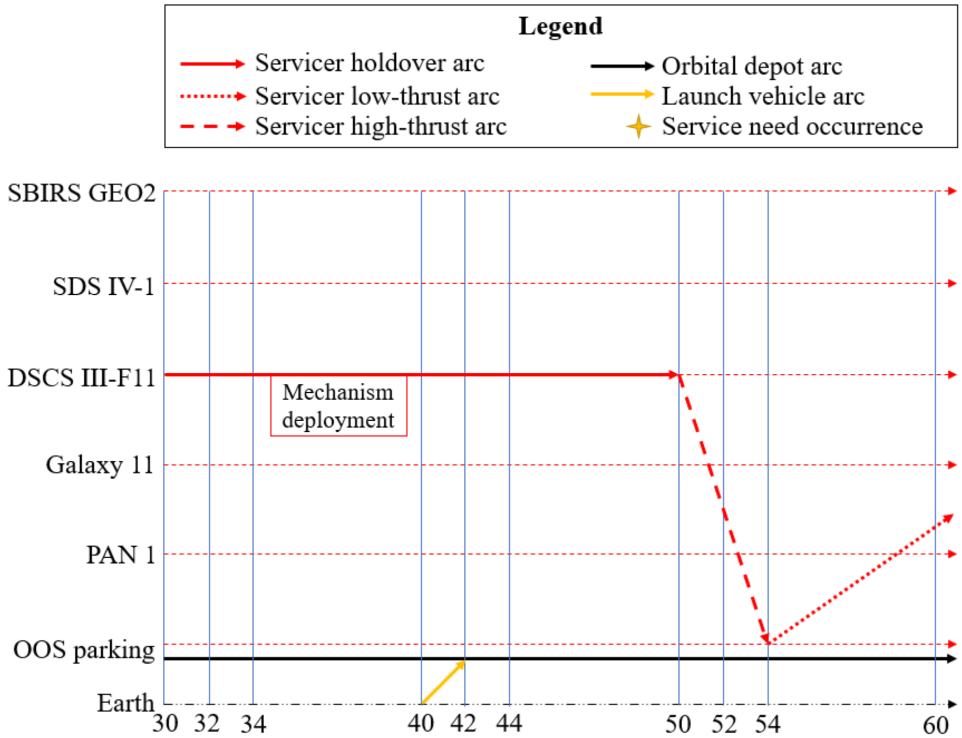

Figure 5.9: Breakdown over the second month (days 30-60) of the operations of an OOS infrastructure consisting of a multimodal versatile servicer and an orbital depot.

For this analysis, 18 simulations are run on an Intel® Core™ i7-9700, 3.00GHz platform with the Gurobi 9 optimizer. The simulation times range between 1 and 129 minutes with an average MILP gap between 0 and 0.13%.

The results for the values of Architecture 1 (i.e., one high-thrust versatile servicer) and Architecture 2 (i.e., four high-thrust specialized servicers) are first presented in Figure 5.11 for the three different levels of service demands. As can be seen on this figure, the monolithic architecture is more valuable than the distributed one over the five-year time horizon for a small OOS market (e.g., 30 satellites). This is due to a higher initial investment and operating cost of the distributed architecture. With a small market, the OOS infrastructures cannot pay back their initial investments in less than five years of business operations. However, as the market grows to a customer base of 71 satellites, the distributed architecture is seen to be catching up with the value of the monolithic architecture. Under this market condition, the monolithic architecture is barely valuable after five years of operations while the

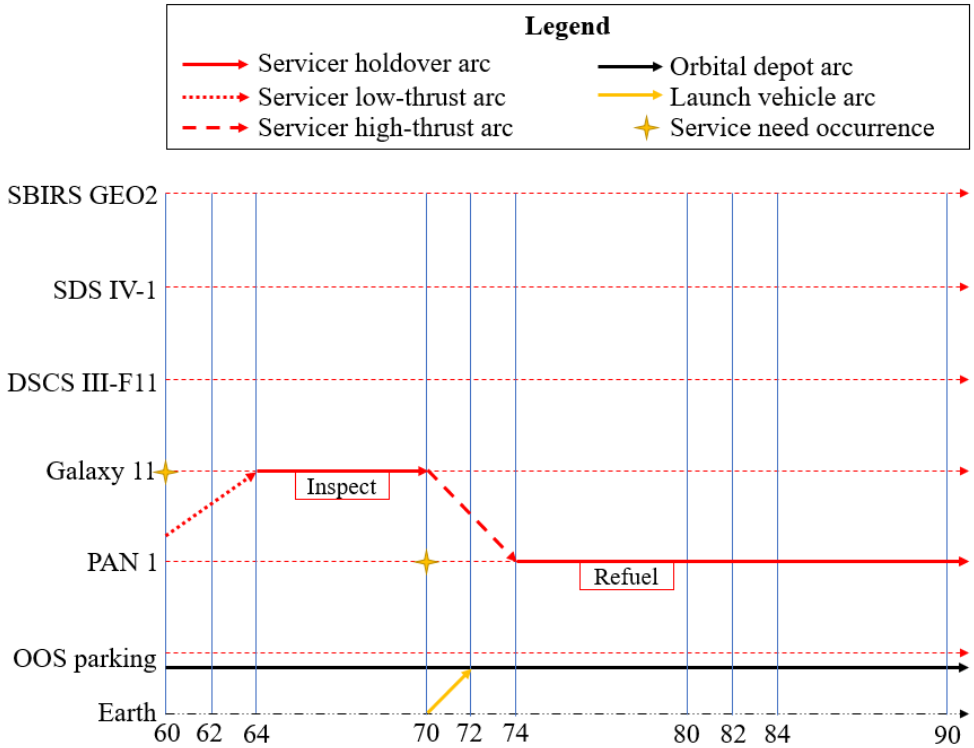

Figure 5.10: Breakdown over the third month (days 60-90) of the operations of an OOS infrastructure consisting of a multimodal versatile servicer and an orbital depot.

distributed architecture is still \$100M away from the breakeven. Finally, for a large market of 142 customer satellites, both architectures become valuable after 2.5 years of operations with the distributed architecture being more profitable than the monolithic one. This is because, unlike the monolithic architecture, the distributed architecture can parallelize the provision of services with four different servicers thus allowing for a larger profitability rate. Essentially, this analysis shows that the distributed architecture has a larger potential in marginal profitability as the market grows than the monolithic architecture. Note that the results presented in Figure 5.11 slightly differ from those presented in chapter 4 because the high-thrust servicers are given the option to fly over four-day trajectories instead of over two-day trajectories only.

A similar analysis is carried out to compare the values of Architecture 3 (i.e., 1 low-thrust versatile servicer) and Architecture 4 (i.e., four low-thrust specialized servicers). The results are presented in Figure 5.12 for the three different levels of service demands. A sim-

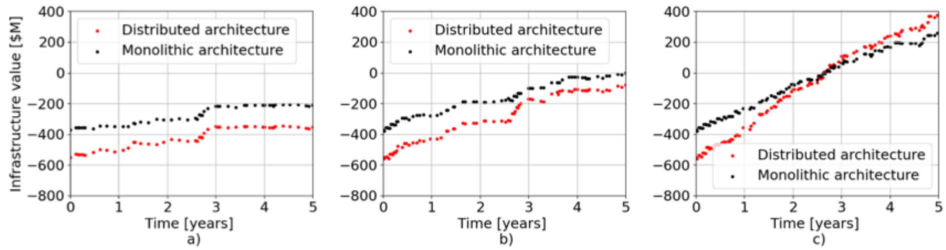

Figure 5.11: Values of the high-thrust monolithic and distributed architectures (Architecture 1 vs Architecture 2) for: a) 30 satellites; b) 71 satellites; c) 142 satellites.

ilar trend as with the high-thrust architectures can be observed on those figures. However, the distributed architecture catches up faster with the value of the monolithic architecture as the market grows. This can be seen in Figure 5.12 with a customer base of 71 satellites: the low-thrust distributed architecture becomes more valuable than its monolithic counterpart after 2.8 years of operations. As justified for the high-thrust architectures, the low-thrust distributed architecture has a greater potential in marginal profitability as the market grows due to the parallelization of the services.

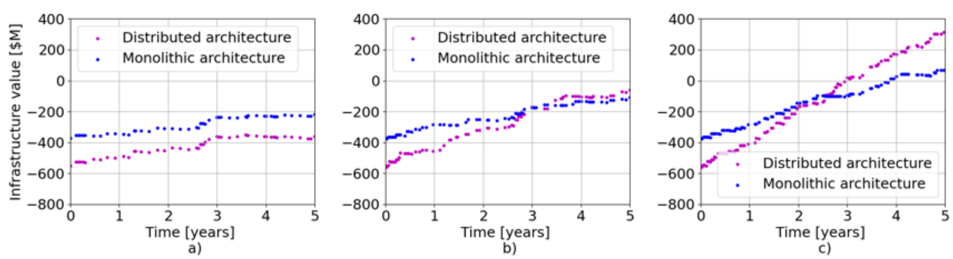

Figure 5.12: Values of the low-thrust monolithic and distributed architectures (Architecture 3 vs Architecture 4) for: a) 30 satellites; b) 71 satellites; c) 142 satellites.

Finally, Architecture 5 (i.e., 1 multimodal versatile servicer) and Architecture 6 (i.e., four multimodal specialized servicers) are compared. The results are presented in Figure 5.13 for the three different levels of service demands. One can observe the same trends as with the high- and low-thrust architectures. Again, the multimodal distributed architecture has a greater potential in marginal profitability as the market grows due to the parallelization of the services.

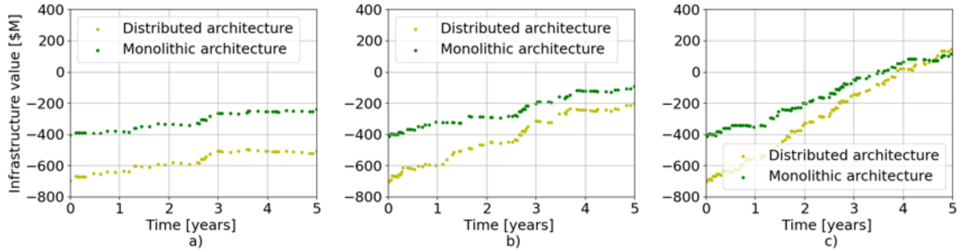

Figure 5.13: Values of the multimodal monolithic and distributed architectures (Architecture 5 vs Architecture 6) for: a) 30 satellites; b) 71 satellites; c) 142 satellites.

Trading propulsion options: high-thrust vs Low-thrust vs multimodal propulsion

This second analysis aims to demonstrate through an example how the proposed OOS logistics framework can be leveraged to inform decision making about the propulsion systems of the servicers. The results presented here are for an assumed large market of 142 customer satellites and are sourced from the 18 simulations introduced in 5.4.1. Note that the analysis which follows is valid for the assumptions made in subsection 5.4.1. Different assumptions can easily be input to the framework, leading to potentially different conclusions from those presented below.

The focus for this example analysis is on whether using multimodal servicers significantly contributes to the value of the OOS infrastructure. Multimodal servicers offer more flexibility to OOS companies through an optimal balance between high-thrust propulsion (i.e., better responsiveness but higher operating cost) and low-thrust propulsion (i.e., slower but cheaper operations). However, this comes at a cost with respect to the high- and low-thrust servicers: a higher initial investment (+$25M), and an added dry mass (+1,000kg), which in turn either increases the operating costs or reduces the revenues available to the servicers.

To give some insight into this important tradeoff, the propulsion system utilization ratio of the multimodal versatile and specialized servicers presented in Figure 5.14 is recorded and analyzed. This metric gives the OOS planners insight into how frequently the multimodal servicers use their high-thrust and low-thrust engines. In Figure 5.14, one can see that the multimodal specialized servicers (i.e., Architecture 6) use their low-thrust engines most of the time. Indeed, the parallelization of services enabled through the use of multiple servicers gives them more time to perform the rendezvous maneuvers. The optimizer thus tends to favor the low-thrust engine over the high-thrust engine due to a lower propellant consumption. Based on this observation, OOS planners may decide to deploy low-thrust specialized servicers instead of multimodal specialized servicers to simplify and reduce the cost of manufacturing and operations. Now take a look into the propulsion system utilization ratio for the multimodal versatile servicer (Architecture 5). As can be seen in Figure 5.14, the servicer uses its high-thrust engine almost as much as its low-thrust engine, which proves the two technologies are similarly valuable for this type of OOS architecture employing a single versatile servicer. Indeed, in this case, the parallelization of services is inexistent since there is only one servicer to address all service needs in a sequential manner. The optimizer thus makes the most of both high- and low-thrust engines to find the optimal balance between servicing responsiveness and operating costs: the more responsive the infrastructure, the more revenues it generates but also the costlier it becomes to operate. Clearly, the tradeoff between the multimodal versatile servicer and its high- and low-thrust counterparts is not as straightforward as in the case of multimodal specialized servicers, and Figure 5.14 alone is not sufficient to properly inform decision making.

To get a better understanding of the tradeoff between the propulsion technologies of the versatile servicer, the problem needs to be considered from a broader perspective. To that end, Figure 5.15 gives the cost breakdown for Architecture 1 (i.e., one high-thrust versatile servicer), Architecture 3 (i.e., one low-thrust versatile servicer), and Architecture 5 (i.e., one multimodal versatile servicer). As a reminder, the value metric is defined as the revenues minus the sum of the initial investments and operational costs. As can be seen in Figure 5.15, employing a high-thrust versatile servicer is significantly more valuable for a large OOS market (i.e., 142 customer satellites) than the other two options. The multimodal versatile servicer is $143M short of the high-thrust servicer in terms of value. This seems counterintuitive as the multimodal servicer can benefit from its low-thrust engine. However, it also is 1,000 kg heavier, by assumption, than the high-thrust servicer due to the additional propulsion system. A heavier multimodal servicer leads to a larger bi-propellant consumption and may not be able to fly along low-thrust transportation arcs with relatively low servicer mass upper bounds. Architecture 3 (i.e., one low-thrust servicer) is the least valuable of the three options investigated here due to low servicing responsiveness and thus lower revenues overall.

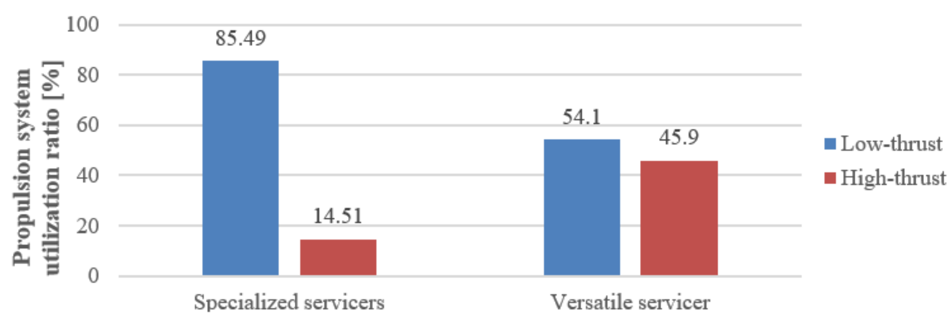

Figure 5.14: Propulsion system utilization ratio of the multimodal versatile servicer (Architecture 5) and multimodal specialized servicers (Architecture 6).

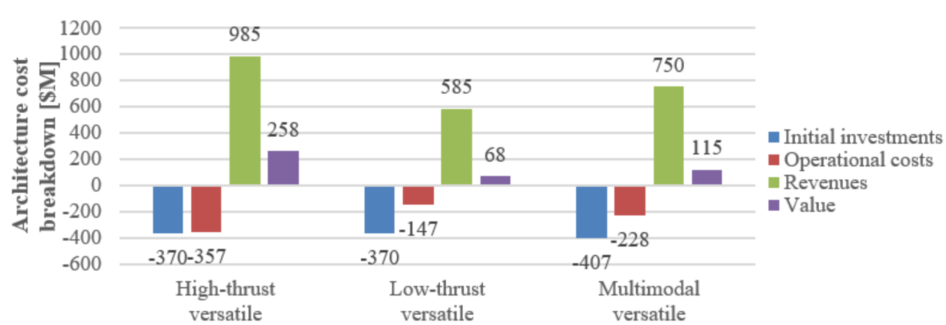

Figure 5.15: Cost breakdown for Architecture 1 (high-thrust versatile servicer), Architecture 3 (low-thrust versatile servicer), and Architecture 5 (multimodal versatile servicer) after five years of operations.

Sensitivity analysis with respect to servicer mass

In this subsection, the sensitivity of the revenues, operating costs, and profits to variations in the mass of the Power Processing Unit (PPU) of the multimodal versatile servicer in

Architecture 5 is analyzed. In this analysis, the initial investments to manufacture and deploy the OOS infrastructure into space are not included in the results to fairly assess the profit-generating performance of the servicer as a function of its dry mass. The long-term strategic planning of Architecture 5 is run for a servicer mass of 3,000kg, 4,000kg, 5,000kg, 6,000kg, an OOS market of 142 customer satellites, and a five-year timeline. Each simulation ran in less than 40 minutes on an Intel® Core™ i7-9700, 3.00GHz platform with the Gurobi 9 optimizer and a MILP gap of 1% a stopping criterion.

Figure 5.16 gives the revenues, the operating costs, and the profits of the architecture which are plotted with the same Y-axis scale to facilitate the comparison between their rates of change. The analysis which follows is valid for the assumptions made in this chapter and should not be used to draw conclusions about the general design of sustainable OOS infrastructures. Instead, OOS planners would need to input their own trustworthy data to the proposed framework, which is highly and easily parameterizable.

The general trend observed on those figures is that decreasing the mass of a servicer has a larger impact on the revenue stream and profits than on the operating costs. For example, after five years of operations, the 3,000 kg servicer has generated about $80 million in excess of those generated by the other three servicers, while the operating costs are similar for all four servicers. Indeed, as the mass of a servicer decreases, the number of feasible rendezvous maneuvers increases as modeled by Equation 5.26 and the concept of servicer mass upper bound. Decreasing the mass of a servicer thus unlocks additional sources of revenues available to the OOS infrastructures by enabling additional feasible trajectories. On the other hand, the revenues generated by the 4,000kg-to-6,000kg servicers are almost identical, which shows that there is a servicer mass threshold between 3,000kg and 4,000kg beyond which the subset of satellites accessible to the servicer through feasible maneuvers remains principally unchanged. In other words, the 3,000 kg servicer taps into more revenue sources than the 4,000kg-to-6,000 kg servicers.

Interestingly, the operating costs are almost identical no matter the mass of the servicer.

Indeed, a lighter servicer will provide more services and thus will need to fly more often than a heavier servicer. However, flying a lighter servicer will require less propellant overall per flight than a heavier servicer.

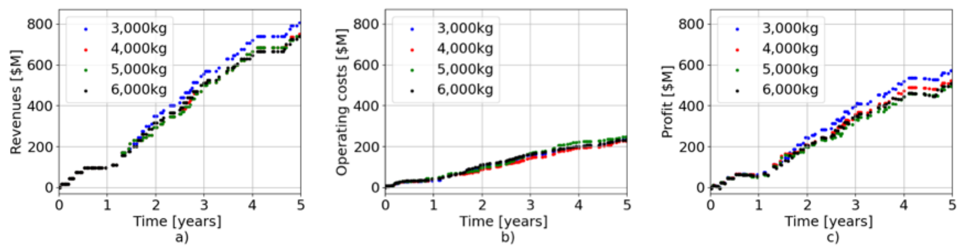

Figure 5.16: a) Revenues; b) Operating costs; and c) profit of Architecture 5 for four different masses of the multimodal versatile servicer.

Discussions

The analyses presented in this chapter do not aim to provide definitive answers regarding the design and operations of OOS infrastructures. Instead, they intend to show the generality of the proposed approach and its ability to perform an automatic tradeoff between the high- and low-thrust propulsion technologies. This method could be used to investigate the long-term value of simpler near-term infrastructures and could easily be modified to assess when and how a given OOS infrastructure should be upgraded in response to a dynamically evolving OOS market.

These analyses are typical examples of how OOS designers could use the proposed framework to effectively inform decision management for the long term. This can be done by running sensitivity analyses over various servicer and depot designs, operational schemes, and even over the nature and price tags of the commodities needed to support the operations of OOS infrastructures. For example, how would the results presented in this section change if a depot is not deployed at all, or its resupply by the launch vehicle is not as frequent as assumed in this chapter (i.e., 30 days)? Wouldn't low-thrust servicers become more valuable than their high-thrust and multimodal counterparts? Is there a situation where the multimodal versatile servicers are more advantageous than high-thrust versatile ones? And what if, as lunar water extraction becomes mainstream, servicers are designed and deployed with water-based low-thrust engines? Xenon gas is an expensive commodity, but water extracted from the Moon may bring in a whole new perspective on the design and operations of OOS infrastructures. As the cislunar economy develops and technology options diversify, OOS planners will need a tool to take a wide variety of parameters into account and come up with competitive and efficient business models.

5.5 Summary

This chapter generalizes the state-of-the-art OOS logistics method presented in chapter 4 by incorporating the automatic tradeoff between the high- and low-thrust propulsion systems of the servicers. The proposed OOS logistics framework is capable of modeling and simulating complex sustainable OOS infrastructures that involve all kinds of depot and servicer models. More specifically, this chapter develops three generic models of servicers: high-thrust-only servicers, low-thrust-only servicers, and multimodal servicers, which embed both high- and low-thrust engines. In addition, the concept of trajectory plugins is introduced that allow OOS planners to build their own high- and low-thrust trajectory models, effortlessly interface them with the framework, and test them as alternative operational strategies for the servicers. The developed piecewise linear approximation model can convert the nonlinear mass transformation model into the MILP-based OOS logistics optimization. Once the trajectory plugins and servicer models are defined, the framework leverages Mixed-Integer Linear Programming and the Rolling Horizon approach to solve the optimal short-term operational scheduling and long-term strategic planning of OOS infrastructures under various states of the OOS market.

The proposed OOS logistics framework makes two fundamental assumptions: (1) the customer satellites are distributed over a common circular orbit; and (2) the orbital depots are staged along the same circular orbit as the fleet of customer satellites. Because GEO servicing falls well within these assumptions, the framework is used to run two different case studies related to this new industry. First, the operations of a multimodal versatile servicer and an orbital depot over a 90-day period are optimized, and the optimizer is shown to be capable of selecting for each rendezvous maneuver the best propulsion technology and trajectory to maximize the short-term profits. The second case study deals with the long-term strategic planning of six different OOS architectures under three different OOS market conditions. The proposed method can effectively support long-term decision making regarding the architectural paradigm of OOS infrastructures (e.g., monolithic vs. distributed) and the propulsion systems to adopt for the servicers.

This chapter is based on the following publication:

T. Sarton du Jonchay, H. Chen, M. Isaji, Y. Shimane, and K. Ho, "On-Orbit Servicing Optimization Framework with High- and Low-Thrust Propulsion Tradeoff," Journal of Spacecraft and Rockets, Vol. 59, No. 1, 2022

CHAPTER 6
ON-ORBIT SERVICING GENERALIZED TO THE MULTI-ORBIT CASE

6.1 Scope

Despite the value of the state-of-the-art OOS optimization frameworks proposed in chapter 4 and chapter 5, the customer satellites and orbital depots are modeled as orbiting the Earth along a single circular orbit. Thus, the nodes of the space network are fixed relative to one another, simplifying the implementation of the rolling horizon procedure and the formulation of the OOS operations as a MILP problem. However, for such a tool to be truly useful to the OOS industry, decision makers must be able to model and simulate customer satellites and orbital depots located on different orbits of various shapes.

The state-of-the-art network-based space logistics frameworks do not consider the relative motion between the nodes of the space network. This means that the cost of transportation between any two nodes is constant throughout the duration of the mission. This is not satisfactory in a multi-orbit OOS setting as different relative positions between two satellites can lead to completely different delta V's. In order to address that gap, this chapter generalizes the OOS framework developed in chapter 4 to the multi-orbit case by tracking the relative motion of the nodes of the network at each time step during the simulation of the OOS operations. This is done by (1) assigning a unique set of orbital elements to each node of the space network; (2) keeping track of the simulation time to properly propagate the orbital elements; and (3) inputting the orbital elements propagated over time into a high-thrust trajectory optimization routine interfaced with the OOS logistics framework to accurately compute the time-varying costs of the network arcs.

6.2 Methods

In this section, the state-of-the-art network-based OOS logistics optimization method is generalized to the multi-orbit case by modeling the relative motion of the nodes of the space network. First, an overview of the static and dynamic networks is given and notion of transportation arc is re-interpreted. Then, the description of how the simulations account for the relative motion of the network nodes over time is given. Finally, the high-thrust trajectory optimization routines interfaced with the framework is presented.

6.2.1 Network and transportation arcs

The operations of the OOS infrastructures are modeled as a MILP problem cast over a network of nodes connected with directed arcs. To rigorously define the mathematical formulation associated with the OOS logistics, a two-step procedure is followed. A static network which represents the state of the network at some given time is first constructed. The static network is then expanded at predefined time steps to create the dynamic network. The dynamic network thus captures the operations of the simulated OOS infrastructure over space and time. Such operations include the flights of the servicers over transportation arcs to visit satellites and provide them with services.

The static network considered in the generalized OOS framework is given in Figure 6.1. Since the nodes of the network are distributed across several orbits of various shapes in this work, the static network in Figure 6.1 is for a given epoch t. In the network, three types of nodes are defined. The Earth nodes are the nodes where the commodities are launched into space from spaceports. The customer nodes are the nodes where the customer satellites triggering the service needs are located. The OOS parking nodes are the nodes where the orbital depots, if any, are deployed and where the servicers are staged when idle. The commodities flowing over the arcs of the network are various types of propellant, spare parts, servicer tools, and the actual vehicle. The first three commodity types cannot fly over an arc unless a vehicle is flying over that arc as well. In addition, between any two nodes, one arc is defined per vehicle type including launch vehicles, orbital depots, and servicers. Finally, seven different service needs are modeled such as refueling and repair, which are of either two natures: deterministic (i.e., re-planned on a regular basis like refueling) and random (i.e., which cannot be predicted like repair).

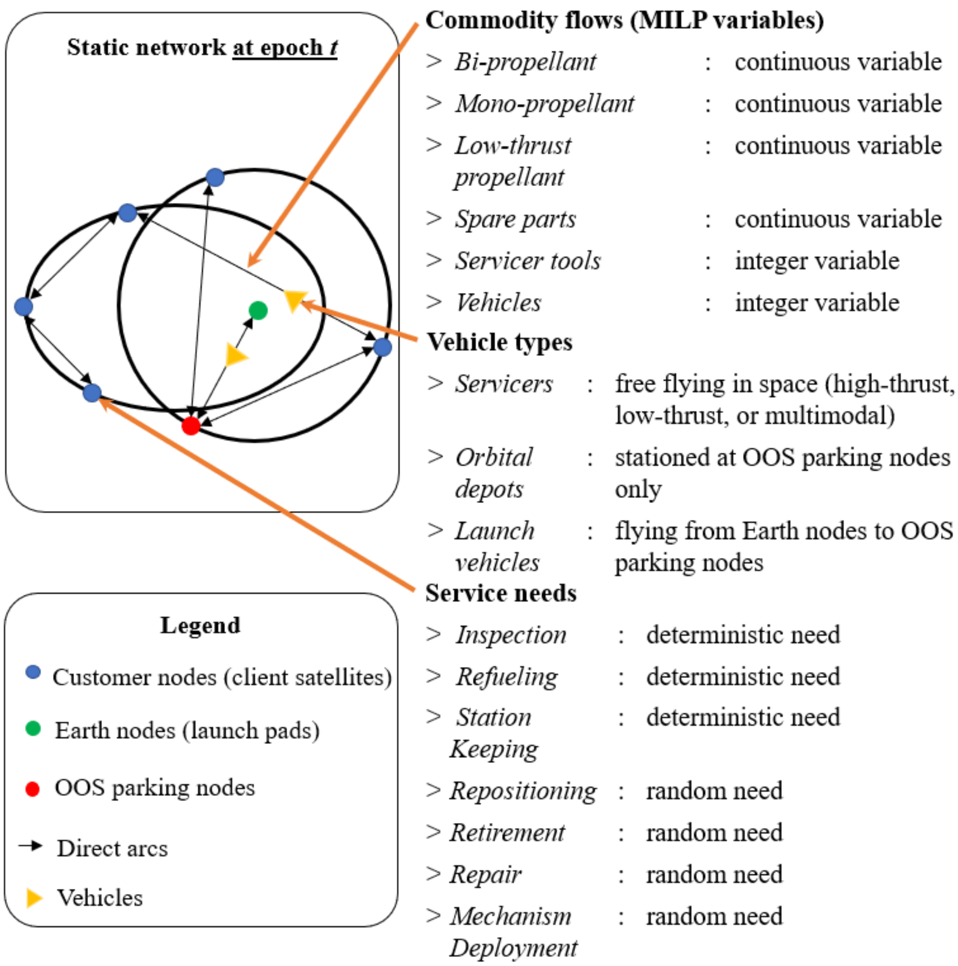

Figure 6.1: OOS static network: state of the network at some epoch t; modified from chapter 4.

Once the static network is defined, it is expanded over time at discrete dates. This is illustrated in Figure 6.2 with a simple three-node network. The dynamic network exhibits a periodic structure with period T. As illustrated in Figure 6.2, the network is replicated at each period and at two additional time steps per period to allow for short-duration servicer flights. The static network is not expanded at every time step to make the MILP problem solvable in a reasonable time. Note that the nodes of the space network change their position at each time step as illustrated by the zoomed-in static networks at $t = 0$ and $t = 40$. How the change in relative positions of the network nodes is accounted for in practice will be discussed in the next subsection.

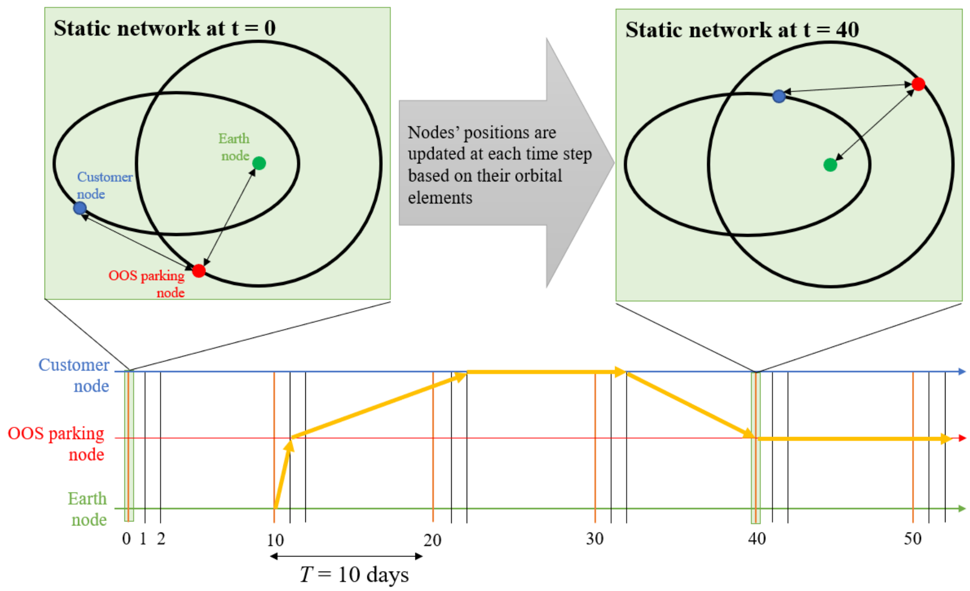

Figure 6.2: Dynamic network used to model the OOS operations over space and time. The network has a period of 10 days. (The horizontal axis is graduated in days.)

Figure 6.2 also illustrates with bold yellow arrows the notional path of a servicer launched from Earth to provide a service at a customer node. In the previous OOS frameworks, simple phasing maneuvers were used for the servicers' transfers between the network nodes, due to a lack of relative motion between the nodes. This means that a servicer was leaving its departure node right at the start of the flight opportunity (represented by the tails of the transportation arcs in Figure 6.2) and reaching its arrival node right at the end of the flight opportunity (represented by the heads of the transportation arcs in Figure 6.2). However, the servicers may be allowed initial and final coasting phases at the departure and arrival nodes, respectively, to save up propellant and make their operations optimal. This is especially important as orbits of various shapes and orientations are now considered. Figure 4.3 illustrates that concept by breaking down the transportation arcs into three phases illustrated with the bold yellow arrows: the initial coasting phase, the actual orbital transfer, and the final coasting phase. The black dashed arrows represent the traditional depictions of transportation arcs.

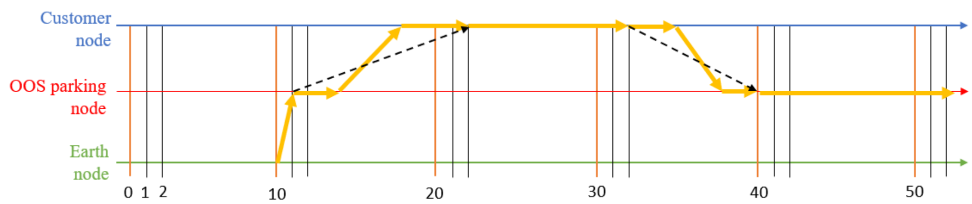

Figure 6.3: Re-interpretation of the notion of transportation arcs in the OOS dynamic network.

6.2.2 Relative dynamics of the network nodes

Now that the dynamic network is re-defined for the multi-orbit OOS logistics problem, the relative motion of the network nodes must be accounted for within the MILP formulation. Before describing how relative motion is accounted for in the simulations, explaining the workflow of the OOS simulations should precede and is illustrated in Figure 6.4. The rolling horizon procedure consists in optimizing successive models to solve dynamic planning problems in uncertain environments. In the context of OOS in this chapter, the optimization of a MILP model is performed over some time interval, called a planning horizon (PH). The PH includes a finite set of service needs that the OOS infrastructure decides whether to provide or not. Once the optimization is over, the state of the OOS infrastructure and customer fleet associated with the end of the control horizon (CH) is saved. The CH is time interval encompassing the first few time steps of the PH. Using the saved states and new service needs occurring over the next PH, the parameters of a new MILP model are then computed, and the next PH optimization is started.

The orbital elements of the network nodes are not updated when the MILP problem is being solved. Indeed, as illustrated in Figure 6.4, once the parameters of a MILP are

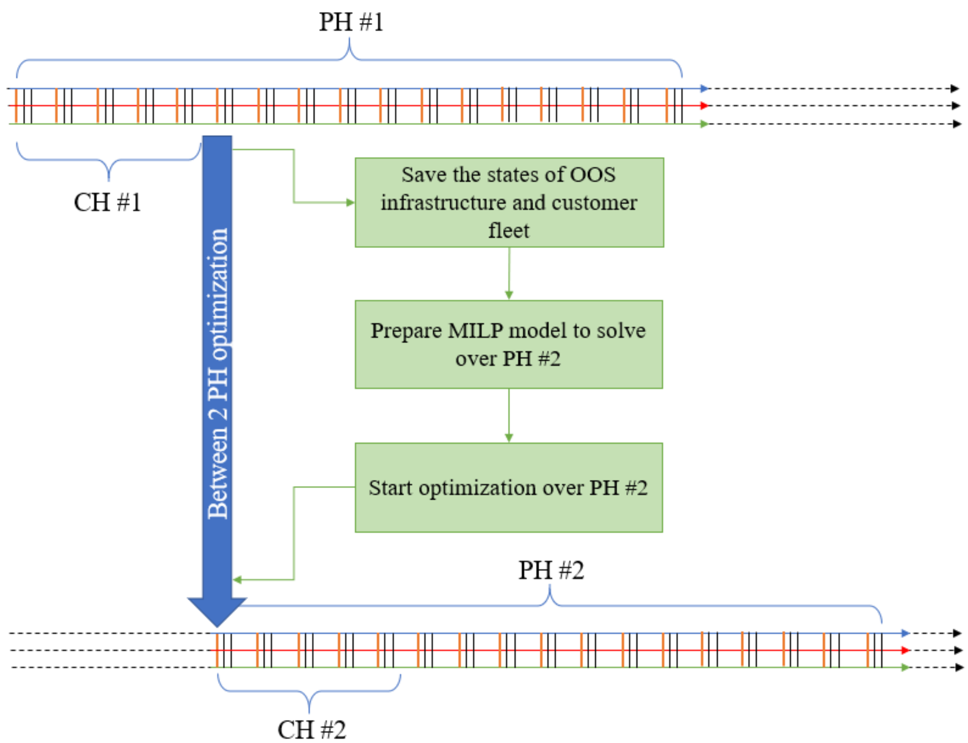

Figure 6.4: Procedure computing the new MILP model between two planning horizon optimizations.

computed, they cannot be tampered with during the optimization. This means that the orbital elements of the network nodes must be propagated while preparing the MILP model and before starting the next PH optimization.

In the OOS frameworks presented in chapter 4 and chapter 5, the nodes are fixed relative to one another, which means that the costs of the transportation arcs are computed only once for all time steps of the PH. In this chapter, however, since the nodes move relative to one another, these costs must be computed at every single time step of the PH. This effectively leads to transportation arc costs that vary over time. Note also that computing these costs at every time step also requires more computational resources compared to the OOS frameworks presented in chapter 4 and chapter 5.

The set of orbital elements is updated in practice by computing the mean anomaly at each time step and solving the corresponding Kepler's equation to find the new true anomaly value. All the other elements remain constant throughout the simulation. The orbital elements of the departure and arrival nodes are then inputted along with additional parameters to trajectory optimization routines that effectively compute the costs of the transportation arcs, i.e., delta V for high-thrust trajectories.

6.2.3 High-thrust trajectory optimization routine

The servicers may be given several time-of-flight options at each time step, for example by flying over a two-day or four-day trajectory. These flight options allow the optimizer to trade between the responsiveness and cost-effectiveness of the servicers' orbital transfers. However, as illustrated in Figure 6.3, the servicers are also allowed to coast along the initial and/or final orbit(s), which means that two transportation arcs with different times of flight may lead to the same trajectory and thus the same transportation cost. For example, if a servicer is given the options to fly to some node along a two-day or a four-day trajectory, the computed trajectories for both options may be identical with the actual orbital transfer happening within two days of the beginning of the transportation arc. Provided this happens, two different transportation arcs with identical cost values would be represented in the network.

To avoid the above scenario, bounds on the end date of the actual orbital transfer are defined as inputs to the trajectory optimization routines. Referring to the previous example of a servicer with two-day and four-day trajectory options, the lower and upper bounds associated with the two-day transportation arc would be 0 and 2, meaning that the start and end dates of the actual orbital transfer would occur between t and t+2, where t is the value of the time step at which the transportation arc starts. For the four-day transportation case, the bounds would be 2 and 4, meaning that the end of the actual orbital transfer would occur between t+2 and t+4. This example is generalizable to any set of transportation arcs of different lengths.

The high-thrust trajectory optimization routine is constructed using Pygmo [47] and is solved using Ipopt [48]. In the remaining of this subsection are defined: $\mathbf{X}$ the decision vector containing the variables to optimize; τ the duration of the combined initial coasting phase and actual orbital transfer; τ_{wait} the duration of the initial coasting phase; r_{wait} the ratio of τ_{wait} over τ; $\tau_{transfer} = \tau - \tau_{wait}$ the duration of the actual orbital transfer; $\mathbf{S}_1$ and $\mathbf{S}_2$ the sets of orbital elements of the departure and arrival nodes at the beginning of the transportation arc; $\mathbf{r}$ and $\mathbf{v}$ the position and velocity vectors; and τ_{lower} and τ_{upper} the bounds defined at the beginning of this subsection.

The high-thrust scenario considered in this chapter involves an optimal two-impulse transfer within an allowable range of time of flight. The formulation builds on a multi-revolution Lambert's problem [49] and consists of solving an optimization problem to minimize the sum of the maneuver costs $\Delta V1$ and $\Delta V2$ of the two impulses. The decision vector $\mathbf{X}$ consists of the total transfer time τ and the wait time ratio r_{wait}. The duration of the orbital transfer $\tau_{transfer}$ is used as the time of flight required to solve Lambert's problem. The evaluation of the two-impulse transfer is as follows:

1. Compute wait time $\tau_{wait} = \tau r_{wait}$
2. Propagate initial orbital elements $\mathbf{S}_1$ by τ_{wait} to obtain $\mathbf{S}_1^+$;
3. Propagate final orbital elements $\mathbf{S}_2$ by τ to obtain $\mathbf{S}_2^+$;
4. Compute Lambert transfer time $\tau_{transfer} = \tau\,(1 - r_{wait})$;
5. Convert $\mathbf{S}_1^+$ and $\mathbf{S}_2^+$ to cartesian state-vectors $\mathbf{x}_1^+ = \left[\mathbf{r}_1^+, \mathbf{v}_1^+\right]$ and $\mathbf{x}_2^+ = \left[\mathbf{r}_2^+, \mathbf{v}_2^+\right]$;
6. Solve multi-revolution Lambert problem to $\mathbf{r}_1^+$ and $\mathbf{r}_2^+$ with $\tau_{transfer}$ as the time of flight;
7. Compute the first and second impulse magnitudes $\Delta V1$ and $\Delta V2$.

An additional constraint preventing the spacecraft from getting too close to Earth is enforced by comparing a safety radius, r_{safety}, with the value of the maneuver's perigee r_p obtained from the Lambert solver. A constraint is also placed on the duration τ to be within the time-of-flight bounds τ_{lower} and τ_{upper}. Thus, the high-thrust optimization problem is given by:

$$\begin{cases} \min_X (\Delta V_1 + \Delta V_2) \\ \text{where } X = [\tau, r_{\text{wait}}] \\ \text{subject to } r_{\text{safety}} - r_{\text{p}} \leq 0 \\ \text{and } \tau_{\text{lower}} \leq \tau \leq \tau_{\text{upper}} \end{cases} \tag{6.1}$$

The high-thrust trajectory optimization routine runs in 10s of milliseconds on the Intel® Core™ i7-9700, 3.00GHz platform.

6.3 Applications

This section demonstrates the value of the generalized OOS framework on the use cases of short-term operational scheduling and long-term strategic planning of sustainable servicing infrastructures involving high-thrust servicers.

To demonstrate the value of the generalized OOS framework with high-thrust servicers, this section is divided into three subsections. First, the assumptions related to the fleet of customer satellites and the OOS infrastructure are given and the two case studies presented. Then is demonstrated the value of the multi-orbit OOS framework to the use cases of short-term operational scheduling (case study 1) and long-term strategic planning (case study 2) in the last two subsections.

6.3.1 Assumptions and Scenarios

In this subsection are given the assumptions related to the customer satellites and OOS infrastructure before describing the case studies' scenarios.

Customer Fleet Assumptions

The data related to the deterministic and random service needs are given in Table 5.2 and Table 5.3. For detailed definitions of the service needs, see chapter 3. Note that the given data are for one customer satellite; by increasing the number of customer satellites in the simulations, the service need rates of the entire fleet of customer satellites increase.

The customer satellites considered in this chapter are GEO satellites with different inclinations and right ascension of the ascending node (RAAN). The orbital elements were retrieved from the Two-Line Elements stored on the CelesTrak database for the "Weather," "Active geosynchronous," "GNSS," and "Space & Earth science" satellites [50]. Of this compiled database of satellites only those in GEO and with orbital inclinations ranging between 0 and 15 degrees were selected as the pool of customer satellites for the simulations presented in this chapter.

OOS infrastructure assumptions

The four notional servicer tools given in Table 3.5 have an assumed cost of $100,000 and an assumed mass of 100 kg. The other commodities considered in the case studies are the spares (assumed price tag: $1,000/kg), bipropellant for the servicers (price tag for Monomethyl Hydrazine: $180/kg), and monopropellant (price tag for Hydrazine: $230/kg). Monopropellant is used to refuel customer satellites for their station keeping operations.

An orbital depot is assumed pre-deployed on a GEO equatorial orbit. The depot is assumed to consume its own monopropellant at a rate of 0.14 kg/day for station keeping [43]. The manufacturing and operating costs of the depot are assumed to be $200M and $13,000/day, respectively.

The launch vehicle used for this analysis is based on a Falcon 9 launcher with an assumed maximum payload capacity of 8,300 kg. A launcher is assumed to be available every 30 days for resupplying the depot. The mass-specific launch cost is assumed to be

Table 6.1: Assumptions related to the high-thrust servicers; modified from Table 5.4.

	Versatile	**Specialized 1 (S1)**	**Specialized 2 (S2)**	**Specialized 3 (S3)**	**Specialized 4 (S4)**
Tools	T1, T2, T3, T4	T1	T2	T3	T4
Dry mass [kg]	3,000	2,000	2,000	2,000	2,000
Propellant capacity [kg]	1,000	1,000	1,000	1,000	1,000
Manufacturing cost [$M]	75	50	50	50	50
Operating cost [$/day]	13,000	13,000	13,000	13,000	13,000
Propellant type	Bi-propellant	Bi-propellant	Bi-propellant	Bi-propellant	Bi-propellant
Specific Impulse [s]	316	316	316	316	316
Flight durations [days]	2, 4, 10, 14	2, 4, 10, 14	2, 4, 10, 14	2, 4, 10, 14	2, 4, 10, 14

$11,300/kg.

Two different high-thrust servicer designs are simulated in this chapter based on the number of tools they are integrated with: one versatile servicer, and four specialized servicers. The versatile servicer can provide all seven defined services, whereas the specialized servicers can only provide the services for which their tools are suited. The detailed assumptions are given in Table 6.1. The baseline dry mass for the high-thrust versatile servicer is taken from [17]. This baseline mass is decreased for the high-thrust specialized servicers, which are assumed to be less capable and smaller in size than their versatile counterpart.

Case studies' scenarios

Two case studies are considered in this chapter: the short-term operational scheduling of four high-thrust specialized servicers; and the long-term strategic planning of two different OOS architectures. Four different market conditions are defined by considering 30, 57, 114, or 228 satellites.

The first case study aims to demonstrate that the OOS framework developed in this chapter can simulate complex servicing infrastructures with customer satellites distributed across orbits at various inclinations. The regular scheduling of the short-term operations of OOS infrastructures will be essential to account for random demand (e.g., repair need)

Table 6.2: Scenarios definition for the short-term operational scheduling case study.

Servicers	4 high-thrust specialized servicers
Depot	1 depot pre-deployed in GEO equatorial orbit
Planning horizon	100 days
Customer fleet	228 customer satellites
Initial conditions	- servicer S1 flying to Amos-3 satellite
	- servicer S2 docked to orbital depot
	- Servicer S3 providing a service to HYLAS-1 satellite
	- Servicer S4 docked to the orbital depot

and remain competitive. In this first case study, it is assumed that at the beginning of the simulation, servicer S3 is already providing a repair service at the satellite HYLAS-1, while servicer S1 is flying to the AMOS-3 satellite. The framework is run for a single planning horizon of the rolling horizon procedure. This scenario is summarized in Table 6.2.

The second case study aims to compare the long-term value (i.e., initial investments subtracted from the profits) of two different OOS architectures. This is done by running the OOS logistics framework for a five-year time horizon, while leveraging the rolling horizon procedure embedded in the framework to account for the random service needs. For this case study, the value of two different OOS architectures dedicated to a fleet of 30, 57, 114, and 228 customer satellites is assessed. The simulated architectures are: (1) a monolithic architecture involving an orbital depot and a single high-thrust versatile servicer; and (2) a distributed architecture involving an orbital depot and four high-thrust specialized servicers. As in the first case study, an orbital depot is pre-deployed on a GEO equatorial orbit.

6.3.2 Case study 1: short-term operational scheduling

The OOS simulation framework is used in this mode to re-plan the operations of any servicing infrastructure already deployed in space. Its users would input the initial conditions of the infrastructure and run it for the next few months of operations with an updated set of service needs. The output would be a breakdown of the operations of the servicers and their interactions with the orbital depots and customer satellites.

The scenario for case study 1 was run on an Intel® Core™ i7-9700, 3.00GHz platform with the Gurobi 9 optimizer. The solution was reached in less than 45 seconds with a gap of 1% stopping criterion. Figure 6.5 and Figure 6.6 illustrate the optimal operations of the four high-thrust specialized servicers. The inclination of each satellite is indicated within brackets.

Note that only eight satellites are represented in Figure 6.5 and Figure 6.6 out of the 228. This is because only these eight satellites experience a service need over that planning horizon. The remaining 220 satellites are not included in the network to minimize the size of the MILP model to be solved.

Also, note the ability of the optimizer in choosing the time-of-flight options that maximize the profits of the OOS infrastructure over the planning horizon. For example, servicer S3 flies along a ten-day trajectory from HYLAS-1 to the parking orbit and then flies along a four-day trajectory to ABS-7 to minimize the penalty fees resulting from servicing delays.

6.3.3 Case study 2: long-term strategic planning

The purpose of this subsection is to show that the multi-orbit OOS framework can be used to perform general market analyses involving complex OOS infrastructures and customer satellites distributed across different orbits with various shapes and orientations. This use case is very useful to new entrants in the OOS market who desire to design and deploy a competitive OOS infrastructure.

The scenario of case study 2 was run on an Intel® Core™ i7-9700, 3.00GHz platform with the Gurobi 9 optimizer. The longest simulation (four high-thrust specialized servicers and 228 customer satellites) took a little less than 2.5 days to run, most of the computation time being dedicated to the preparation of the MILP problems rather than to actually solve them. Figure 6.7 through Figure 6.10 illustrate the value (i.e., initial investments subtracted from profits) of the monolithic and distributed infrastructures for four different

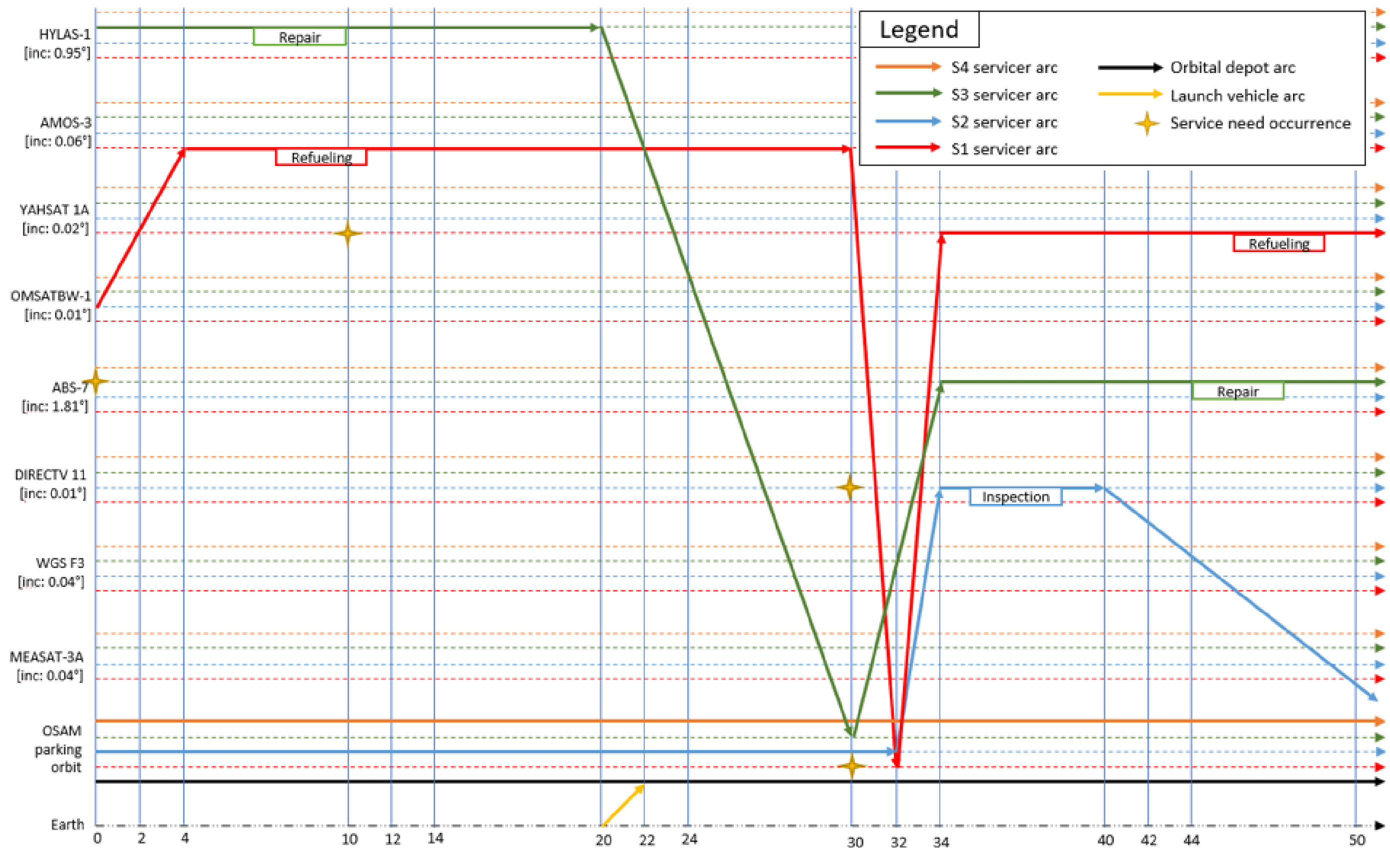

Figure 6.5: Breakdown of the operations of the four high-thrust specialized servicers over the initial 50 days of the planning horizon.

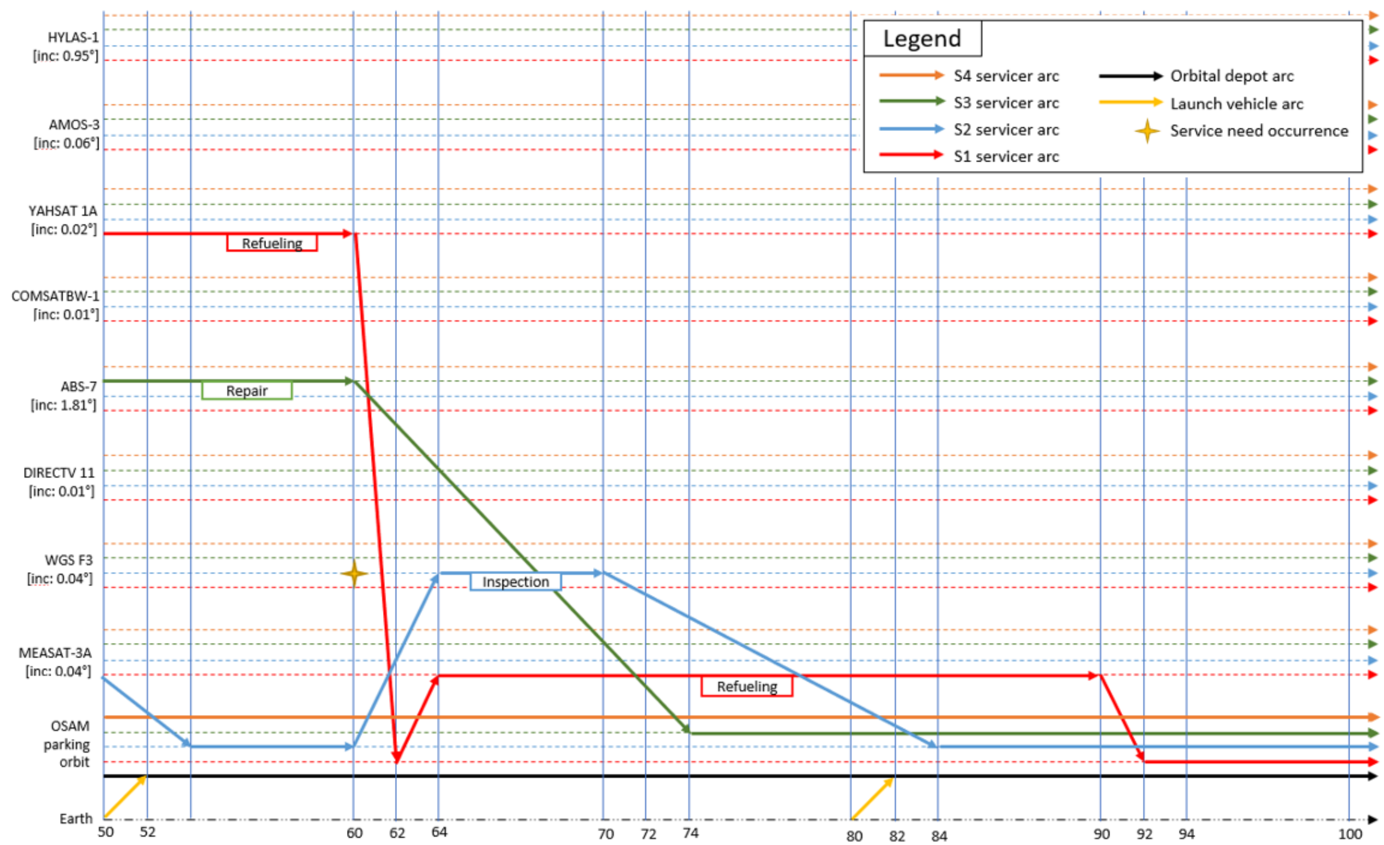

Figure 6.6: Breakdown of the operations of the four high-thrust specialized servicers over the final 50 days of the planning horizon.

market conditions.

From the figures, one can see that naturally, as the number of customer satellites increases, the values of both the monolithic and distributed architectures increase. However, although the values of both architectures evolve at the same rate for 30, 57, and 114 customer satellites, there is a threshold satellite number beyond which the distributed architecture becomes more valuable than the monolithic architecture. For example, for a market condition of 228 satellites, the distributed architecture becomes more valuable than the monolithic architecture after about two years of operations. Finally, for the market conditions of 30, 57 and 114 satellites, neither the monolithic architecture nor the distributed one reaches the breakeven point before the five-year mark. However, for a market condition of 228 satellites, the monolithic architecture reaches the breakeven point after 3.5 years of operations while the distributed architecture reaches it earlier, after 2.8 years of operations.

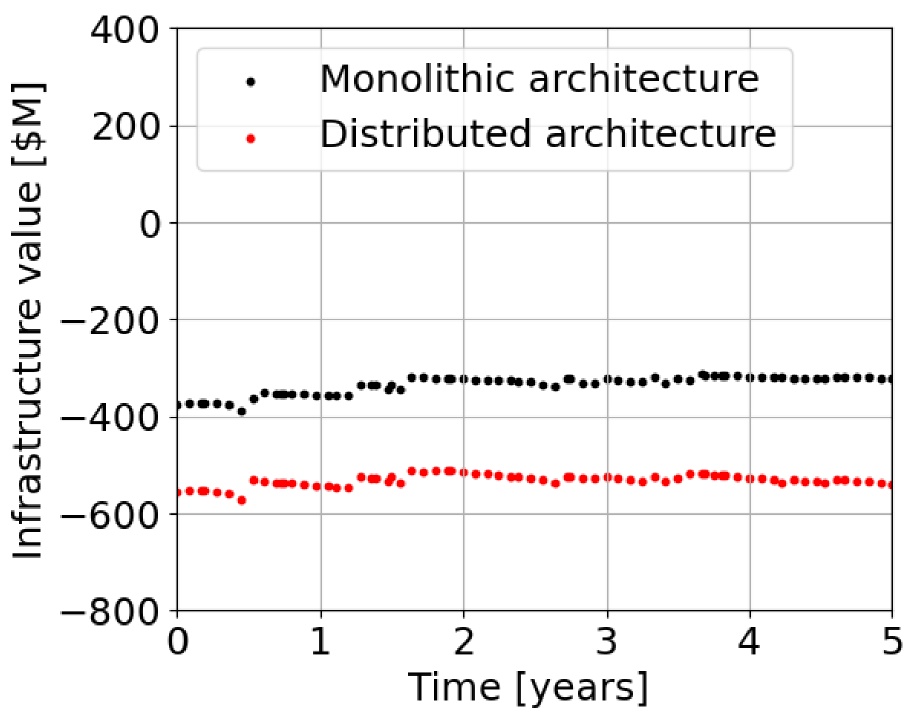

Figure 6.7: Values of the high-thrust monolithic and distributed architectures for a market condition of 30 customer satellites.

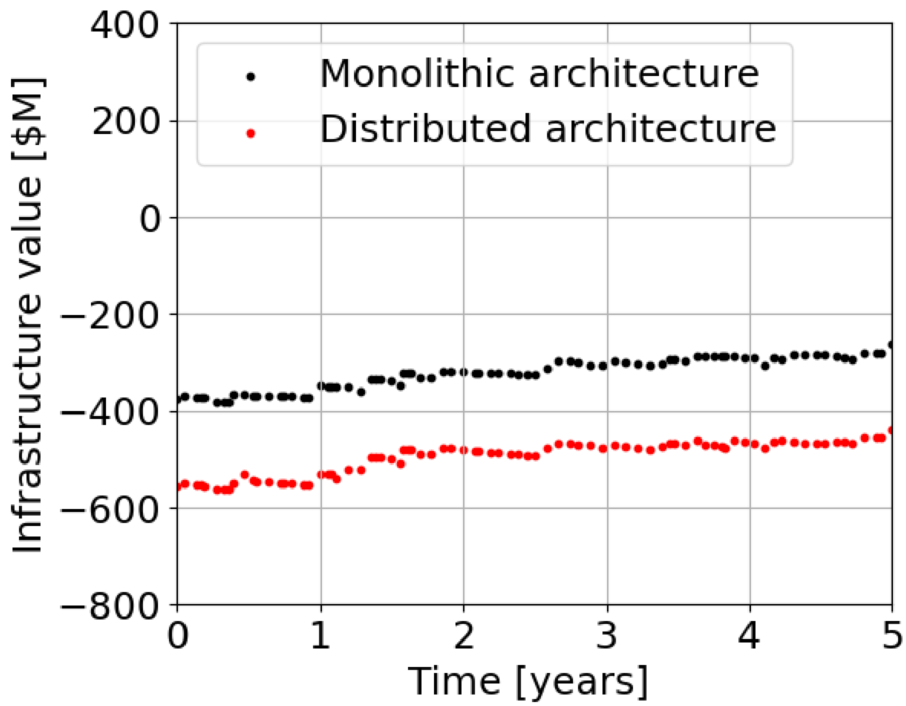

Figure 6.8: Values of the high-thrust monolithic and distributed architectures for a market condition of 57 customer satellites.

6.4 Summary

This chapter generalizes the state-of-the-art network-based OOS logistics method to the multi-orbit case with high-thrust servicers. This is done by keeping track of the simulation time to properly propagate the orbital elements of the nodes in the space network. The updated orbital elements are then inputted to a high-thrust trajectory optimization routines to accurately compute the costs of the transportation arcs in the dynamic network.

Two case studies are presented to demonstrate the capability of the developed OOS framework in optimizing the operations of complex sustainable servicing infrastructures dedicated to satellites distributed across orbits of various orientations and shapes. The first case study demonstrates the short-term operational scheduling of a high-thrust distributed architecture, whereas the second case study demonstrates the long-term strategic planning of high-thrust monolithic and distributed architectures.

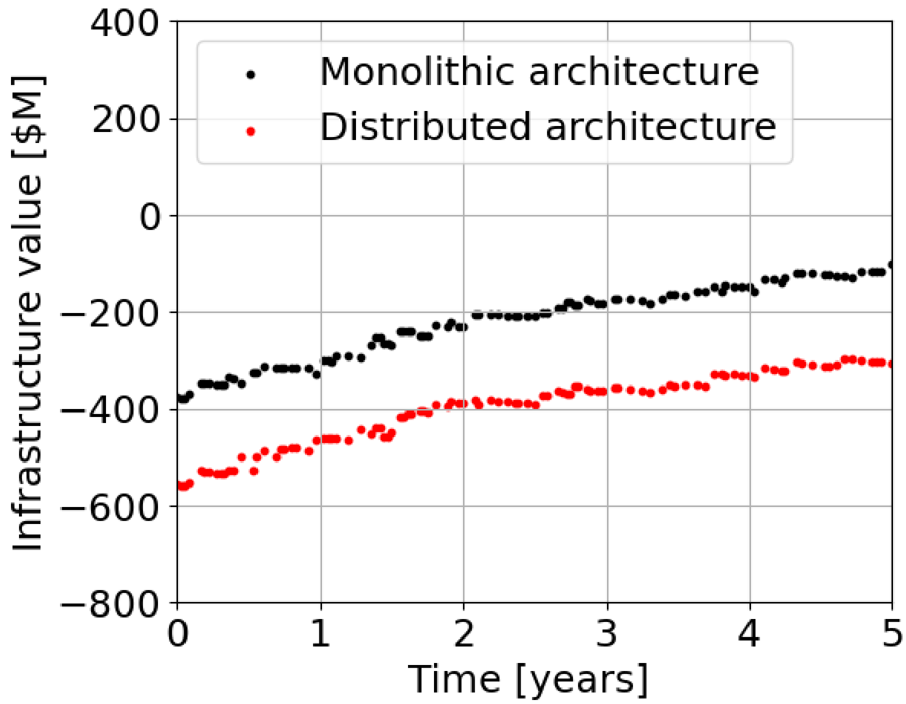

Figure 6.9: Values of the high-thrust monolithic and distributed architectures for a market condition of 114 customer satellites.

This chapter is based on the following conference paper:

Sarton du Jonchay, T., Shimane, Y., Isaji, M., Chen, H. and Ho, K., "On-Orbit Servicing Logistics Framework Generalized to the Multi-Orbit Case," AAS/AIAA Astrodynamics Specialist Conference, Big Sky, Virtual, Aug. 2021.

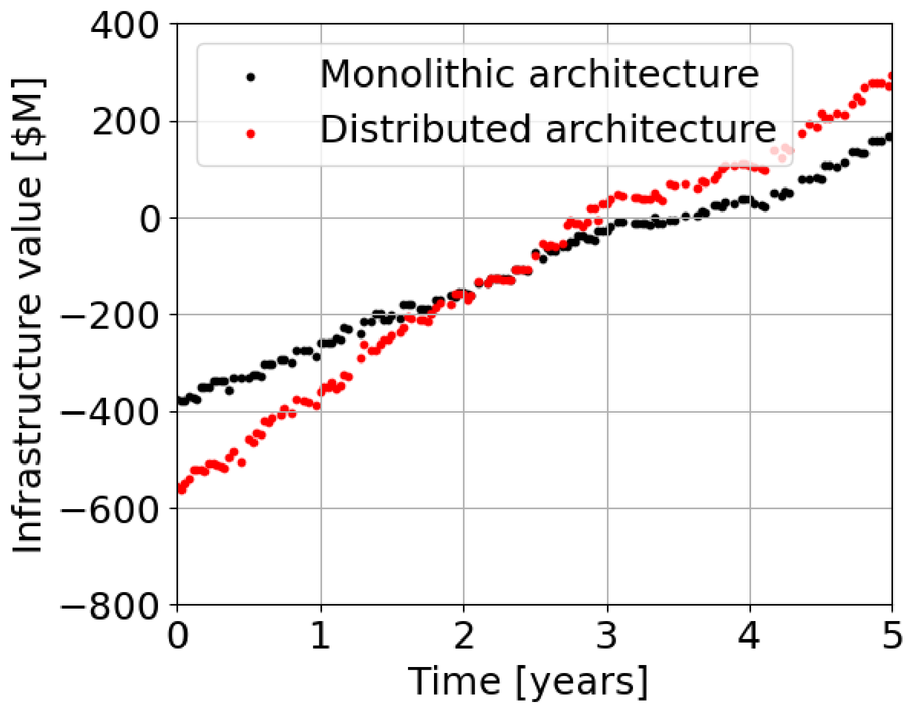

Figure 6.10: Values of the high-thrust monolithic and distributed architectures for a market condition of 228 customer satellites.

CHAPTER 7
CONCLUSION

The contributions of this dissertation are summarized below:

1. Framework for modeling and operations of OOS infrastructures under service demand uncertainties (chapter 4)

    - This chapter generalizes the Traveling Salesman Problem by allowing for the modeling of a supply chain of various commodities - including but not limited to vehicles, propellant, spares, tools – and interacting elements, e.g., servicers getting refueled at depots and re-supply vehicles resupplying depots with spare satellite components. The Traveling Salesman Problem only models the flow of a single commodity: the servicing spacecraft.
    - The Time-Expanded Generalized Commodity Network Flow (TE-GMCNF) model developed in state-of-the-art space logistics studies is generalized in this chapter by constraining vehicles, namely servicers in the context of OOS, to remain at a certain node of the OOS network for some pre-defined time. This is done through the definition of two additional sets of variables – the servicer dispatch variables and the service assignment variable – and additional operational constraints.
    - The uncertainties in service demand are accounted for through the Rolling Horizon (RH) method which results in the re-planning of the operations of the OOS infrastructures whenever a random service need occurs. Unlike the RH approach typically encountered in the literature, the RH method implemented in this OOS planning framework involves Control Horizons that are allowed to differ in length to match the occurrence dates of random service needs.

- This chapter offers a tool that mission planners can use for both the short-term operational scheduling of existing OOS infrastructures and the long-term strategic planning of future OOS infrastructures.

2. OOS framework with high-thrust and low-thrust propulsion tradeoff (chapter 5)

   - The TE-GMCNF formulation developed in chapter 4 is generalized so mission planners can effortlessly integrate user-defined trajectories to the framework. In particular, two indices are added to the formulation so the optimizer knows over which type of trajectory and for how long the servicers must fly. This allows mission planners to trade various trajectory options all within the same tool.
   - Chapter 5 also relaxes the assumption made in chapter 4 that the servicers are equipped with high-thrust engines. In chapter 5's framework, high-thrust, low-thrust and/or multimodal servicers, (i.e., equipped with both high- and low-thrust engines) can be modeled, allowing for convenient tradeoffs between different servicer designs all within the same tool.

3. OOS framework generalized to the multi-orbit case (chapter 6)

   - Chapter 6 leverages the generalized TE-GMCNF formulation developed in chapter 5 to model the operations of high-thrust servicers dedicated to the servicing of GEO satellites distributed across orbits of various shapes and orientations. This is done by computing the relative dynamics of the nodes of the network over the simulation time before inputting the time-varying orbital elements into thrust-thrust trajectory optimization routines. This results in a time-expanded network with time-varying costs over the transportation arcs, unlike the networks defined in the frameworks developed in chapter 4 and chapter 5.

REFERENCES

[1] S. Eenma, *Investment perspectives: On-orbit satellite servicing markets continue to evolve. ISS national laboratory*, https://www.issnationallab.org/blog/investment-perspectives-on-orbit-satellite-servicing-markets-continue-to-evolve/, [Accessed 2/24/2020].

[2] J. Foust, *Rethinking satellite servicing. The Space Review*, https://www.thespacereview.com/article/3653/1, [Accessed 2/24/2020].

[3] C. Henry, *Geostationary satellite orders bouncing back. SpaceNews*, https://spacenews.com/geostationary-satellite-orders-bouncing-back/, [Accessed 1/26/2021].

[4] *OSAM-1: On-orbit servicing, assembly and manufacturing. NASA*, https://nexis.gsfc.nasa.gov/OSAM-1.html, [Accessed 1/26/2021].

[5] J. Parrish, *Robotic servicing of geosynchronous satellites. DARPA*, https://www.darpa.mil/program/robotic-servicing-of-geosynchronous-satellites, [Accessed 1/26/2021].

[6] SpaceLogistics, https://www.northropgrumman.com/space/space-logistics-services/, [Accessed 1/26/2021].

[7] *NSR press releases. Northern Sky Research*, https://www.nsr.com/nsr-report-forecasts-4-5-billion-in-cumulative-revenues-from-in-orbit-satellite-services-by-2028/, [Accessed 1/26/2021].

[8] J. Foust, *OrbitFab and Benchmark Space Systems to partner on in-space refueling technologies. SpaceNews*, https://spacenews.com/orbit-fab-and-benchmark-space-systems-to-partner-on-in-space-refueling-technologies/, [Accessed 7/26/2021]., Feb. 23, 2021.

[9] S. Erwin, *Lockheed Martins to upgrade gps satellites for in-orbit servicing. SpaceNews*, https://spacenews.com/lockheed-martin-to-upgrade-gps-satellites-for-in-orbit-servicing/, [Accessed 7/26/2021]., Feb. 26, 2021.

[10] *Aerojet rocketdyne propulsion help enable new satellite servicing market. Aerojet Rocketdyne*, https://www.rocket.com/media/news-features/ar-propulsion-helps-enable-new-satellite-servicing-market, [Accessed 1/26/2021].

[11] *Xenon acquisition strategies for high-power electric propulsion nasa missions. NASA*, https://ntrs.nasa.gov/api/citations/20150023079/downloads/20150023079.pdf, [Accessed 1/26/2021].

[12] *Convert prices and calculate cost of materials and compounds. Aqua-calc*, https://www.aqua-calc.com/calculate/materials-price, [Accessed 1/26/2021].

[13] D. Werner, *Orbit fab to launch first fuel tanker in 2021 with spaceflight. SpaceNews*, https://spacenews.com/orbit-fab-to-launch-with-spaceflight/, [Accessed 1/31/2021].

[14] *Monomethylhydrazine*, http://www.astronautix.com/m/mmh.html, [Accessed 1/26/2021].

[15] W. Yao, X. Chen, Y. Huang, and M. van Tooren, "On-orbit servicing system assessment and optimization methods based on lifecycle simulation under mixed aleatory and epistemic uncertainties," *Acta Astronautica*, vol. 87, pp. 8–13, 2013.

[16] S. Zhao, P. Gurfil, and J. Zhang, "Optimal servicing of geostationary satellites considering earth's triaxiality and lunisolar effects," *Journal of Guidance, Control, and Dynamics*, vol. 39, pp. 2219–2231, 2016.

[17] A. W. Verstraete, D. Anderson, S. Louis, N. M., and J. Hudson, "Geosynchronous earth orbit robotic servicer mission design," *Journal of Spacecraft and Rockets*, vol. 55, pp. 1444–1452, 2018.

[18] T. Sarton du Jonchay and K. Ho, "Quantification of the responsiveness of on-orbit servicing infrastructure for modularized earth orbiting platforms," *Acta Astronautica*, vol. 132, pp. 192–203, 2017.

[19] K. Ho, D. Wang H., P. A., T. Sarton du Jonchay, and K. Tomita, "Semi-analytical model for design and analysis of on-orbit servicing architecture," *Journal of Spacecraft and Rockets*, vol. 57, no. 6, 2020.

[20] P. Sear and K. Ho, "Impact evaluation of in-space additive manufacturing and recycling technologies for on-orbit servicing," *Journal of Spacecraft and Rockets*, vol. 56, no. 6, pp. 1498–1508, 2018.

[21] J. S. Hudson and D. Kolosa, "Versatile on-orbit servicing mission design in geosynchronous earth orbit," *Journal of Spacecraft and Rockets*, vol. 57, no. 4, pp. 844–850, 2020.

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

[23] T. Sarton du Jonchay, "Modeling and simulation of permanent on-orbit servicing infrastructures dedicated to modularized earth-orbiting platforms," *Master's Thesis, University of Illinois at Urbana-Champaign*, 2017.

[24] T. H. Matos de Carvalho and J. Kingston, "Establishing a framework to explore the servicer-client relationship in on-orbit servicing," *Acta Astronautica*, vol. 153, pp. 109–121, 2018.

[25] K. K. Galabova and O. L. de Weck, "Economic case for the retirement of geosynchronous communication satellites via space tugs," *Acta Astronautica*, vol. 58, no. 9, pp. 458–498, 2016.

[26] C. Taylor, M. Song, D. Klabjan, O. L. de Weck, and D. Simchi-Levi, "A mathematical model for interplanetary logistics," *Logistics Spectrum*, vol. 41, no. 1, pp. 23–33, 2007.

[27] P. Grogan, H. Yue, and O. L. de Weck, "Space logistics modeling and simulation analysis using spacenet: Four application case," *AIAA SPACE 2011 Conference & Exposition*, vol. AIAA Paper 2011-7346, Sept. 2011.

[28] D. Arney and A. Wilhite, "Modeling space system architectures with graph theory," *Journal of Spacecraft and Rockets*, vol. 51, no. 5, pp. 1413–1429, 2014.

[29] T. Ishimatsu, O. L. de Weck, J. Hoffman, Y. Ohkami, and R. Shishko, "Generalized multicommodity network flow model for the earth-moon–mars logistics system," *Journal of Spacecraft and Rockets*, vol. 53, no. 1, pp. 25–38, 2016.

[30] K. Ho, O. L. de Weck, J. Hoffman, and R. Shishko, "Dynamic modeling and optimization for space logistics using time-expanded networks," *Acta Astronautica*, vol. 105, no. 2, pp. 428–443, 2014.

[31] K. Ho, O. L. de Weck, J. Hoffman, and R. Shishko, "Campaign-level dynamic network modelling for spaceflight logistics for the flexible path concept," *Acta Astronautica*, vol. 123, pp. 51–61, 2016.

[32] H. Chen and K. Ho, "Integrated space logistics mission planning and spacecraft design with mixed-integer nonlinear programming," *Journal of Spacecraft and Rockets*, vol. 55, no. 2, pp. 365–381, 2018.

[33] H. Chen, H. Lee, and K. Ho, "Space transportation system and mission planning for regular interplanetary missions," *Journal of Spacecraft and Rockets*, vol. 56, no. 1, pp. 12–20, 2019.

[34] H. Chen, T. Sarton du Jonchay, L. Hou, and K. Ho, "Integrated in-situ resource utilization system design and logistics for mars exploration," *Acta Astronautica*, vol. 170, pp. 80–92, 2020.

[35] B. Jagannatha and K. Ho, "Event-driven network model for space mission optimization with high-thrust and low-thrust spacecraft," *Journal of Spacecraft and Rockets*, vol. 57, no. 3, pp. 446–463, 2020.

[36] J. P. Vielma, S. Ahmed, and G. Nemhauser, "Mixed-integer models for nonseparable piecewise linear optimization: Unifying framework and extensions," *Operations Research*, vol. 58, no. 2, pp. 303–315, Oct. 2009.

[37] S. Sethi and G. Sorger, "A theory of rolling horizon decision making," *Annals of Operations Research*, vol. 29, pp. 387–415, 1991.

[38] J. Silvente, G. Kopanos, E. Pistikopoulos, and A. Espuña, "A rolling horizon optimization framework for the simultaneous energy supply and demand planning in microgrids," *Applied Energy*, vol. 155, pp. 485–501, 2015.

[39] Q. Dishan, H. Chuan, L. Jin, and M. Manhao, "A dynamic scheduling method of earth-observing satellites by employing rolling horizon strategy," *Scientific World Journal*, vol. 2013, 2013.

[40] *UCS satellite database. Union of Concerned Scientists*, www.ucsusa.org/resources/satellite-database, [Accessed 7/17/2020].

[41] *Premiere for europe: Jules verne refuels the iss. The European Space Agency*, http://www.esa.int/Science_Exploration/Human_and_Robotic_Exploration/ATV/Premiere_for_Europe_Jules_Verne_refuels_the_ISS, [Accessed: 9/28/2020].

[42] C. Quilty and A. Moeller, "January 2018 satellite & space monthly review," *Quilty Analytics*, Feb. 2018.

[43] B. R. Sullivan and D. L. Akin, "Satellite servicing opportunities in geosynchronous orbit," *AIAA SPACE 2012 Conference and Exposition*, no. AIAA Paper 2012-5261, 2012.

[44] E. M. L. Beale and J. A. Tomlin, "Special facilities in a general mathematical programming system for non-convex problems using ordered sets of variables," *Operational Research*, vol. 69, pp. 447–454, 1970.

[45] *Session 6: Analytical approximations for low thrust maneuver. Massachussetts Institute of Technology*, https://ocw.mit.edu/courses/aeronautics-and-astronautics/16-522-space-propulsion-spring-2015/lecture-notes/MIT16_522S15_Lecture6.pdf, [Accessed 1/31/2021].

[46] *In-space propulsion data sheet. Aerojet Rocketdyne*, https://www.rocket.com/sites/default/files/documents/In-Space%20Data%20Sheets%204.8.20.pdf.

[47] D. Izzo, "PyGMO and PyKEP: Open source tools for massively parallel optimization in astrodynamics (the case of interplanetary trajectory optimization)," *Fifth International Conf. Astrodynam. Tools and Techniques*, 2012.

[48] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Math. Program.*, vol. 106, no. 1, pp. 25–57, 2006.

[49] D. Izzo, "Revisiting lambert's problem," *Celest. Mech. Dyn. Astron.*, vol. 121, no. 1, 2014.

[50] *Two-line element sets current data. CelesTrak*, https://www.celestrak.com/NORAD/elements/, [Accessed 7/28/2021].