AAS 21-590

ON-ORBIT SERVICING LOGISTICS FRAMEWORK GENERALIZED TO THE MULTI-ORBIT CASE

Tristan Sarton du Jonchay,* Yuri Shimane,* Masafumi Isaji,* Hao Chen,† and Koki Ho‡

This paper proposes a multi-orbit on-orbit servicing logistics optimization framework capable of planning the operations of sustainable servicing infrastructures with client satellites distributed across different orbits of various shapes. The proposed framework generalizes the state-of-the-art network-based on-orbit servicing logistics optimization method to the multi-orbit case by tracking the relative motion of the network nodes as part of the process of computing the costs of the network arcs. The new framework keeps track of the simulation time in order to propagate the orbital elements of the network nodes over time. The orbital elements are then inputted into high-thrust and low-thrust trajectory optimization routines interfaced with the framework to accurately compute the cost of transportation of the servicers. Finally, a mixed-integer linear program is formulated to model the operations of the servicing infrastructure over the network and over time, whereas the rolling horizon procedure is leveraged to account for the uncertainties in service demand. 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.

INTRODUCTION

On-Orbit Servicing (OOS) is a nascent space-based industry aimed at making the operations and management of Earth-orbiting satellites sustainable. Two space systems central to OOS infrastructures are servicers and orbital depots. Servicers are robotic spacecraft designed to provide a set of services to client satellites, such as refueling and/or repair. Orbital depots are in-space warehouses that store commodities such as spares and/or propellant to support the long-term operations of the servicers and client satellites.

The success of Northrop Grumman's servicers MEV-1 and MEV-2 in the life extension of two Intelsat satellites[1] has led to enhanced optimism in the promises of this new industry both from satellite operators and servicer developers. For instance, Lockheed Martins recently announced that the bus of the next generation of GPS satellites, located in Medium Earth Orbit (MEO), were being redesigned to be serviceable[2], while Astroscale, a company originally created to tackle the challenge of space debris removal, is branching out its activities to Low Earth Orbit (LEO) and Geosynchronous orbit (GEO) servicing as a direct contender to Northrop Grumman's OOS business[3].

* Ph.D. Student, School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA 30332.
† Postdoctoral Fellow, School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA 30332.
‡ Assistant Professor, School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA 30332.

Although the development of servicers started twenty years ago with DARPA and NASA's Orbital Express program[4], the development of orbital depots took off only recently with Orbit Fab, developing and launching the first space gas station in LEO to support prospective client satellites equipped with a specific refueling interface[5].

What the recent media coverage demonstrates is that OOS is deemed to be an industry dedicated to a variety of client satellites distributed across a wide range of orbital regimes, from LEO to MEO and GEO. As the OOS market becomes more and more crowded, OOS companies face critical strategic decisions as to what orbital regime to serve, especially as the supporting infrastructure, such as orbital depots, is being deployed. Should they dedicate their servicers to GEO servicing only, as Northrop Grumman seems to favor? What would be the most profitable range of orbital inclinations to serve? Could they afford to serve customer satellites distributed over orbits with widely different shapes and sizes? Could they serve a wide variety of orbital regimes after deploying their own space gas stations, thus winning additional market shares? These are legitimate questions that will need answers as competition increases among servicing companies.

In order to explore the marketability and competitiveness of their business models, OOS companies need tools to help them make valuable operational and strategic decisions. The past literature has already contributed to important aspects of OOS mission planning. Previous projects specifically focused on the rigorous and accurate analysis and design of high-thrust[6] and low-thrust[7] servicer trajectories using high-fidelity force models. Others analyzed the operations of OOS infrastructures from a scheduling[8] and/or design standpoint[9,10,11] through simulations[10,12,13,14,15], optimization[8,11,16], or a mixture of both[9]. None of these studies, however could model and simulate complex sustainable OOS infrastructures involving as many servicers and orbital depots as desired while simultaneously modeling service demand uncertainties.

To address this gap, References 17 and 18 developed OOS optimization frameworks that model the fleet of customer satellites as a network of nodes that servicers, supported by orbital depots, must visit based on the service needs arising at these satellites. Reference 17 proposed the application of the rolling horizon decision-making procedure to account for the uncertainties related to the service needs, modeled the customer satellites to be along a unique circular orbit, and considered high-thrust servicers only. Reference 18 then proposed to extend the framework developed in Reference 17 with low-thrust servicers and *multimodal* servicers, i.e., servicers equipped with both high- and low-thrust propulsion systems. In both frameworks, important concepts are adapted from past network-based space logistics projects to solve the OOS logistics problem. First, the operations of the OOS infrastructures over the space network are solved as a mixed-integer linear program (MILP), a globally optimal solution method employed in state-of-the-art space logistics frameworks.[19,20,21,22,23,24,25] Then, the concept of *multiarcs*[20,21,22,23,24,25] between the nodes of the network is leveraged to represent the different servicer options as well as the different trajectory options per servicer. For example, an OOS infrastructure may include one high-thrust servicer and one low-thrust servicer, in which case both servicers are assigned their own sets of arcs to travel between the same pairs of nodes in the network. Finally, the piecewise linear approximation of non-linear functions[19,20,26] is used to incorporate the non-linear propellant consumption model of low-thrust servicers into MILP formulations.

Despite the value of the state-of-the-art OOS optimization frameworks proposed in References 17 and 18, the customer satellites and orbital depots are considered to be 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. In reality, 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 ΔV's. In order to address that gap, this paper generalizes the OOS framework proposed in Reference 18 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 high-thrust and low-thrust trajectory optimization routines that are interfaced with the OOS logistics framework to accurately compute the time-varying costs of the network arcs.

Although the developed method can be applied to various OOS concepts, we consider the following OOS concept in this paper 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 such as propellant and spare parts. 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 customer 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 comes back to its orbital staging location for storage, or flies to another customer satellite if need be. The notional services modeled in this paper are inspection, refueling, station keeping, satellite repositioning, repair, mechanism deployment, and retirement. They fall in either of two categories: random (e.g., unplanned service needs such as component failure) or deterministic (e.g., pre-planned service needs such as refueling).

The remaining of the paper is organized as follows. The second section describes in detail how the multi-orbit OOS optimization framework simulates the relative motion of the network nodes and how it accurately computes the costs assigned to the network arcs. The third section demonstrates the framework through the use cases of short-term operational scheduling and long-term strategic planning of sustainable OOS infrastructures. Finally, the fourth section summarizes and concludes this paper.

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, we give an overview of the static and dynamic networks and re-interpret the notion of transportation arc as opposed to the OOS frameworks proposed in References 17 and 18. Then, we describe how the simulations account for the relative motion of the network nodes over time. Finally, the high-thrust and low-thrust trajectory optimization routines interfaced with the framework are presented.

Networks 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, we follow a two-step procedure. We first create a static network which represents the state of the network at some given time. We then expand that static network 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 1. Since the nodes of the network are distributed across several orbits of various shapes in this work, the static network in Figure 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, we define one arc 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). For a detailed definition of the service needs, see Reference 18.

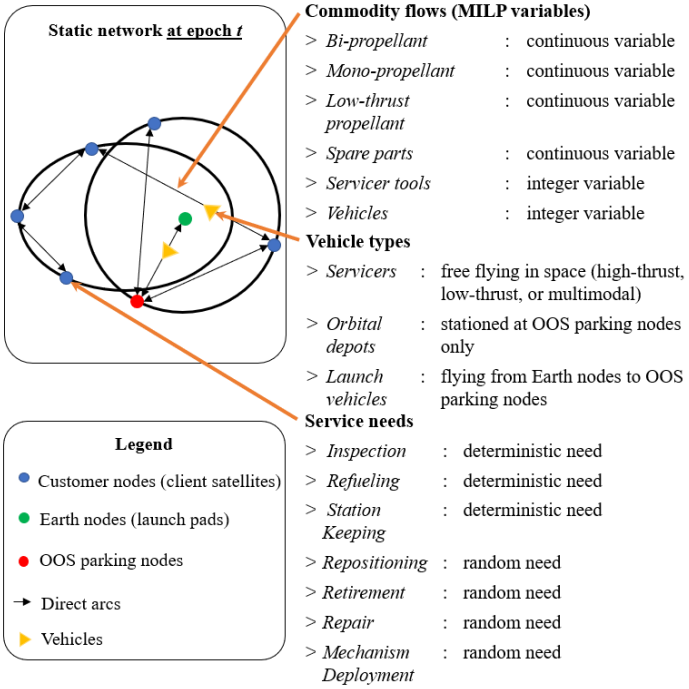

Figure 1. OOS static network: state of the network at some epoch *t*; modified from Reference 18.

Once the static network is defined, we expand it over time at discrete dates. This is illustrated in Figure 2 with a simple 3-node network. The dynamic network exhibits a periodic structure with period *T*. As illustrated in Figure 2, the network is replicated at each period and at two additional time steps per period to allow for short-duration servicer flights. We do not expand the static network 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.

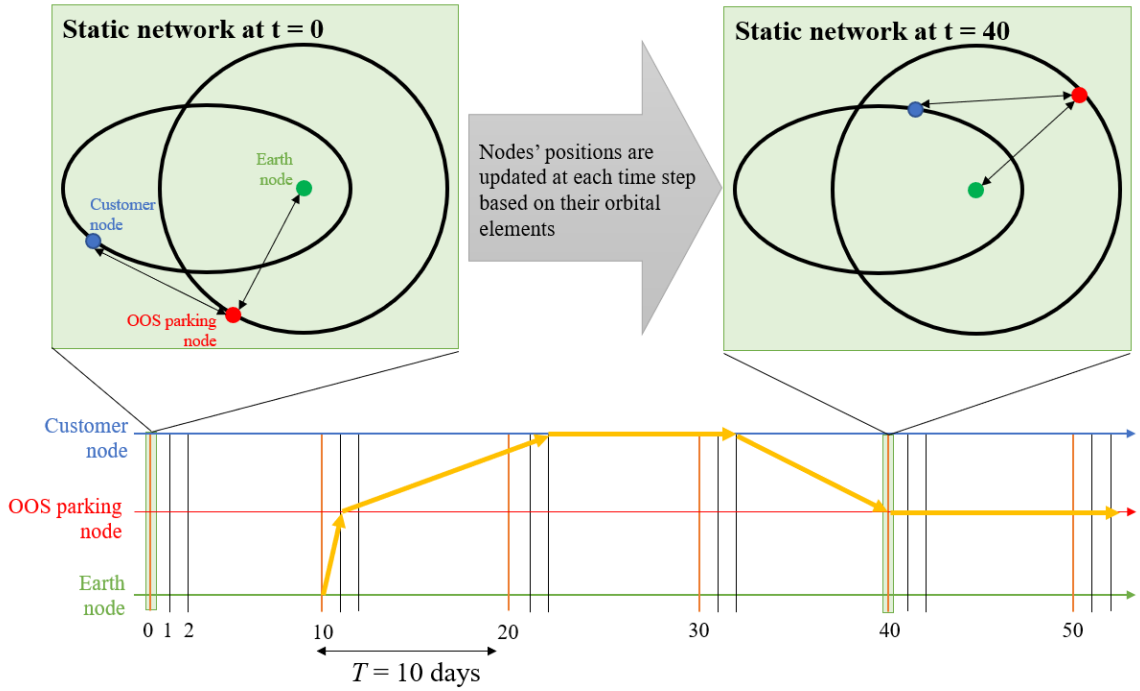

Figure 2. Dynamic network used to model the OOS operations over space and time. The network has a period of 10 days. (The horizontal axis is graduated in days.)

Figure 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 2), and reaching its arrival node right at the end of the flight opportunity (represented by the heads of the transportation arcs in Figure 2). However, in reality, 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 we now consider orbits of various shapes and orientations. Figure 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.

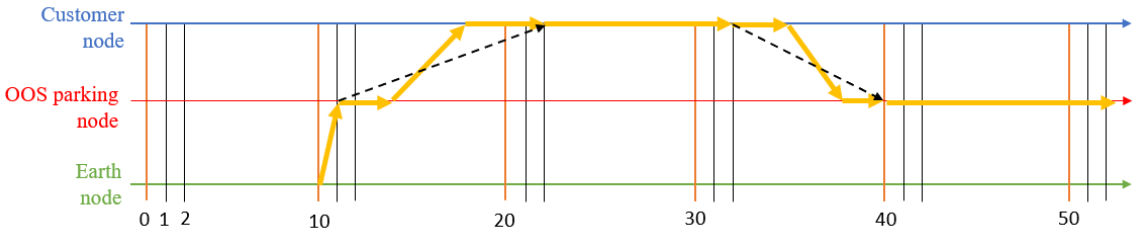

Figure 3. Re-interpretation of the notion of transportation arcs in the OOS dynamic network.

Accounting for the relative motion 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 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 paper, 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, we save the state of the OOS infrastructure and customer fleet associated with the end of the *control horizon* (CH), which is a 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. For more details about the rolling horizon procedure used in this work, see Reference 17.

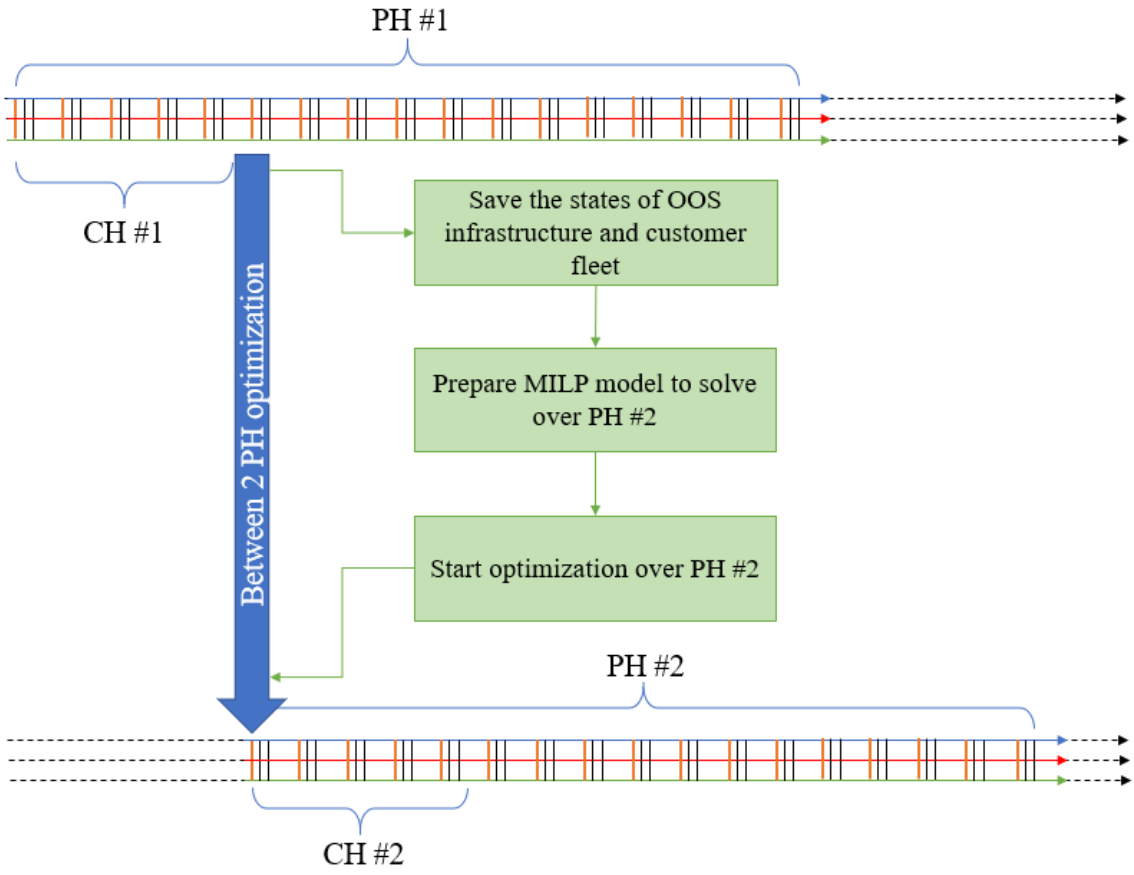

Figure 4. Procedure computing the new MILP model between two planning horizon optimizations.

The orbital elements of the network nodes are not updated when the MILP problem is being solved. Indeed, as illustrated in Figure 4, once the parameters of a MILP are 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 References 17 and 18, 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 the present paper, 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 References 17 and 18.

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., ΔV for high-thrust trajectories, and propellant consumption for low-thrust trajectories.

High-thrust and low-thrust trajectory optimization routines

The servicers may be given several time-of-flight options at each time step, for example by flying over a 2-day or 4-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 3, the servicers are also allowed to coast along the initial and/or final orbit(s), which means that two transportation arcs with different time of flights 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 2-day or a 4-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, we would end up with two different transportation arcs with identical cost values.

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 2-day and 4-day trajectory options, the lower and upper bounds associated with the 2-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 4-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.

For both the high-thrust and low-thrust trajectory routines, the resulting optimization problem is constructed using Pygmo[27] and is solved using Ipopt[28]. In the remaining of this subsection, we define: $\boldsymbol{X}$ 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; $\tau_{\text{transfer}} = \tau - \tau_{\text{wait}}$ the duration of the actual orbital transfer; $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$ the sets of orbital elements of the departure and arrival nodes at the beginning of the transportation arc; $\boldsymbol{r}$ and $\boldsymbol{v}$ the position and velocity vectors; and τ_{lower} and τ_{upper} the bounds defined at the beginning of this subsection.

High-thrust trajectory optimization routine. The high-thrust scenario considered in this paper involves an optimal two-impulse transfer within an allowable range of time of flight. The formulation builds on a multi-revolution Lambert's problem[29] and consists of solving an optimizing problem to minimize the sum of the maneuver costs ΔV_1 and ΔV_2 of the two impulses. The decision vector $\boldsymbol{X}$ consists of the total transfer time τ and the wait time ratio r_{wait}. The duration of the orbital transfer τ_{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 $\tau_{\text{wait}} = \tau r_{\text{wait}}$;
2. Propagate initial orbital elements $\boldsymbol{S}_1$ by τ_{wait} to obtain $\boldsymbol{S}_1^+$;
3. Propagate final orbital elements $\boldsymbol{S}_2$ by τ to obtain $\boldsymbol{S}_2^+$;
4. Compute Lambert transfer time $\tau_{\text{transfer}} = \tau(1 - r_{\text{wait}})$;
5. Convert $\boldsymbol{S}_1^+$ and $\boldsymbol{S}_2^+$ to cartesian state-vectors $\boldsymbol{x}_1^+ = [\boldsymbol{r}_1^+, \boldsymbol{v}_1^+]$ and $\boldsymbol{x}_2^+ = [\boldsymbol{r}_2^+, \boldsymbol{v}_2^+]$;
6. Solve multi-revolution Lambert problem to $\boldsymbol{r}_1^+$ and $\boldsymbol{r}_2^+$ with τ_{transfer} as the time of flight;
7. Compute the first and second impulse magnitudes ΔV_1 and ΔV_2.

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:

$$\begin{cases} \min_{\boldsymbol{X}}(\Delta V_1 + \Delta V_2) \\ \text{where } \boldsymbol{X} = [\tau, r_{\text{wait}}] \\ \text{subject to } r_{\text{safety}} - r_{\text{p}} \leq 0 \\ \text{and } \tau_{\text{lower}} \leq \tau \leq \tau_{\text{upper}} \end{cases} \tag{1}$$

The high-thrust trajectory optimization routine runs in 10s of milliseconds on the Intel® Core™ i7-9700, 3.00GHz platform that we are using to develop the OOS framework.

Low-thrust trajectory optimization routine. The low-thrust scenario is constructed based on the Sims-Flanagan transcription[30], which approximates a low-thrust transfer by considering a series of n impulsive maneuvers. The transfer is divided into a forward and backward leg, propagating the initial states forward and the final states backward. At the halfway point, the difference in cartesian states and spacecraft mass is computed as a residual vector $\boldsymbol{c}_{\text{match-point}}$, which must be driven to zero by the trajectory optimizer. For the two-body dynamics case, the cartesian state propagation may be done using Kepler's equation instead of integrating the equations of motion, which is significantly faster. The mass is propagated using Tsiolkovsky's rocket equation at each impulse. The algorithmic procedure for evaluating a Sims-Flanagan transportation arc is as follows:

1. Propagate initial orbital elements $\boldsymbol{S}_1$ by τ_{wait} to obtain $\boldsymbol{S}_1^+$;
2. Propagate final orbital elements $\boldsymbol{S}_2$ by τ to obtain $\boldsymbol{S}_2^+$;
3. Convert $\boldsymbol{S}_1^+$ and $\boldsymbol{S}_2^+$ to extended state-vectors $\boldsymbol{x}_0 = [x_0, y_0, z_0, v_{x_0}, v_{y_0}, v_{z_0}, m_0]$ and $\boldsymbol{x}_2 = [x_f, y_f, z_f, v_{x_f}, v_{y_f}, v_{z_f}, m_f]$, where m denotes the mass of the servicer;
4. Propagate $\boldsymbol{x}_0$ forward by $\tau_{\text{transfer}}/2$ to obtain $\boldsymbol{x}_0^+$;
5. Propagate $\boldsymbol{x}_f$ backward by $\tau_{\text{transfer}}/2$ to obtain $\boldsymbol{x}_f^-$;
6. Compute the residual at match-point $\boldsymbol{c}_{\text{match-point}} = \boldsymbol{x}_f^- - \boldsymbol{x}_0^+$.

The decision vector $\boldsymbol{X}$ consists of the wait-time τ_{wait}, the transfer time τ_{transfer}, the initial and final masses m_0 and m_f, and $3n$ thrust controls defined at each impulse. The thrust controls consist of the engine throttle ρ, in-plane angle θ, and out-of-plane angle β, where the angles are defined in the local-vertical-local-horizontal frame. A constraint is placed on the duration τ to be within the time-of-flight bounds τ_{lower} and τ_{upper}. The resulting low-thrust optimization problem attempts to maximize the final mass, and is given by:

$$\begin{cases} \min_{\boldsymbol{X}} -m_f \\ \text{where } \boldsymbol{X} = [\tau_{\text{wait}}, \tau_{\text{transfer}}, m_0, m_f, \rho, \theta, \beta] \\ \text{subject to } \tau_{\text{lower}} \leq \tau \leq \tau_{\text{upper}} \\ \text{and } \boldsymbol{c}_{\text{match-point}} = \begin{bmatrix} x_f^- - x_0^+ \\ y_f^- - y_0^+ \\ z_f^- - z_0^+ \\ v_{x_f}^- - v_{x_0}^+ \\ v_{y_f}^- - v_{y_0}^+ \\ v_{z_f}^- - v_{z_0}^+ \\ m_f^- - m_0^+ \end{bmatrix} = \boldsymbol{0} \end{cases} \tag{2}$$

For each low-thrust transportation arc, the trajectory optimization routine is solved for a range of initial mass of the servicer to account for the fact that at the beginning of the maneuver, the amount of commodities (e.g., propellant, spares) in the servicer's propellant tanks and payload bays may assume different values. Thus, each low-thrust transportation arc is assigned several initial mass/final mass pairs. These are then used to define the piecewise linear approximation of the non-linear low-thrust propellant consumption model within the MILP problem. For more details on the piecewise linear approximation of the low-thrust propellant consumption model, see Reference 18.

In addition to the low-thrust propellant consumption model, the *servicer mass upper bound* M^{ub} is computed by the OOS framework for each time step and low-thrust transportation arc of the dynamic network. M^{ub} ensures that overloaded servicers cannot fly over infeasible low-thrust transportation arcs. For more details on the servicer mass upper bound, see Reference 18.

Currently, the low-thrust trajectory optimization routine takes on average 20 to 30 seconds to run on the Intel® Core™ i7-9700, 3.00GHz platform we are using to develop the OOS framework. Since this trajectory routine needs to be solved for several initial servicer masses per transportation arc, preparing each MILP model in a simulation involving low-thrust and/or multimodal servicers can take a significant amount of time.

CASE STUDIES

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. Currently, simulations involving low-thrust and/or multimodal servicers cannot be run in a reasonable amount of time on a typical desktop computer due to the time required to compute the cost of each low-thrust transportation arc. To remediate to this, later work will consider surrogate modeling of the low-thrust trajectory model as well as running the computationally expensive simulations on high-end computing clusters.

To demonstrate the value of the generalized OOS framework with high-thrust servicers, this section is divided into three subsections. First, we give the assumptions related to the fleet of customer satellites and the OOS infrastructure, and present the two case studies. Then, we demonstrate 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.

Assumptions and case studies' scenarios

In this subsection, we first give 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 1 and Table 2, respectively; these are the same assumptions used in the case studies of Reference 18. For detailed definitions of the service needs, see Reference 18. 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 paper 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.[31] 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 paper.

Table 1. Assumptions related to the deterministic service needs (*); from Reference 18.

	Inspection	Refueling	Station Keeping
Revenues [$M]	10 [8]	15 [8]	20
Delay penalty fee [$/day]	5,000 [8]	100,000 [8]	100,000 [8]
Service duration [days]	10 [8]	30 [8]	180 [14]
Service window [days]	30	30	30
Frequency of occurrence [days]	6,310 [16]	2,100 [16]	2,100 [16]

* References are indicated in exponents. Data without reference is assumed. The frequency of occurrence is derived from the data given in Reference 16.

Table 2. Assumptions related to the random service needs (*); from Reference 18.

	Repositioning	Retirement	Repair	Mechanism Deployment
Revenues [$M]	10 [8]	10 [8]	30	25 [8]
Delay penalty fee [$/day]	100,000 [8]	0 [8]	100,000 [8]	100,000 [8]
Service duration [days]	30 [8]	30 [8]	30 [8]	30 [8]
Service window [days]	30	30	30	30
Mean frequency of occurrence [days]	2,520 [16]	2,520 [16]	9,020 [16]	21,050 [16]

* References are indicated in exponents. Data without reference is assumed. The mean frequency of occurrence is derived from the data given in Reference 16.

OOS infrastructure assumptions. The four notional servicer tools given in Table A1 in Appendix 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.[32] 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 $11,300/kg.

Two different high-thrust servicer designs are simulated in this paper based on the number of tools they are integrated with: 1 versatile servicer, and 4 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 3. The baseline dry mass for the high-thrust versatile servicer is taken from Reference 8. 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.

Finally, this paper 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., 2 days).[33] This assumption can be modified depending on the technology performance.

Table 3. Assumptions related to the high-thrust servicers (*); modified from Reference 18.

	Versatile	Specialized 1	Specialized 2	Specialized 3	Specialized 4
Tools	T1, T2, T3, T4	T1	T2	T3	T4
Dry mass [kg]	3,000 [8]	2,000	2,000	2,000	2,000
Propellant capacity [kg]	1,000	1,000	1,000	1,000	1,000
Manufac-turing cost [$M]	75	50 [34]	50 [34]	50 [34]	50 [34]
Operating cost [$/day]	13,000 [15]	13,000 [15]	13,000 [15]	13,000 [15]	13,000 [15]
Propellant type	Bi-propellant	Bi-propel-lant	Bi-propel-lant	Bi-propel-lant	Bi-propel-lant
Specific Im-pulse [s]	316	316	316	316	316
Flight dura-tions [days]	2, 4, 10, 14	2, 4, 10, 14	2, 4, 10, 14	2, 4, 10, 14	2, 4, 10, 14

* References are indicated in exponents. Data without reference is assumed.

Case studies' scenarios. Two case studies are considered in this paper: 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 paper 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) and remain competitive. In this first case study, we assume 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 4.

Table 4. Scenario 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 the orbital depot. - Servicer S3 providing a service to *HYLAS-1* satellite. - Servicer S4 docked to the orbital depot.

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 5-year time horizon, while leveraging the rolling horizon procedure embedded in the framework to account for the random service needs. For this case study, we assess the value of two different OOS architectures dedicated to a fleet of 30, 57, 114, and 228 customer satellites. 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. Finally, although this is not demonstrated in this paper due to long simulation times on a desktop computer, the proposed framework can simulate OOS architectures involving low-thrust and multimodal servicers.

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. Figures 5 and 6 illustrate the optimal operations of the four high-thrust specialized servicers. The inclination of each satellite is indicated within brackets.

Note that only 8 satellites are represented in Figures 5 and 6 out of the 228. This is because only these 8 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 10-day transportation from *HYLAS-1* to the parking orbit and then flies along a 4-day trajectory to *ABS-7* to minimize the penalty fees resulting from servicing delays.

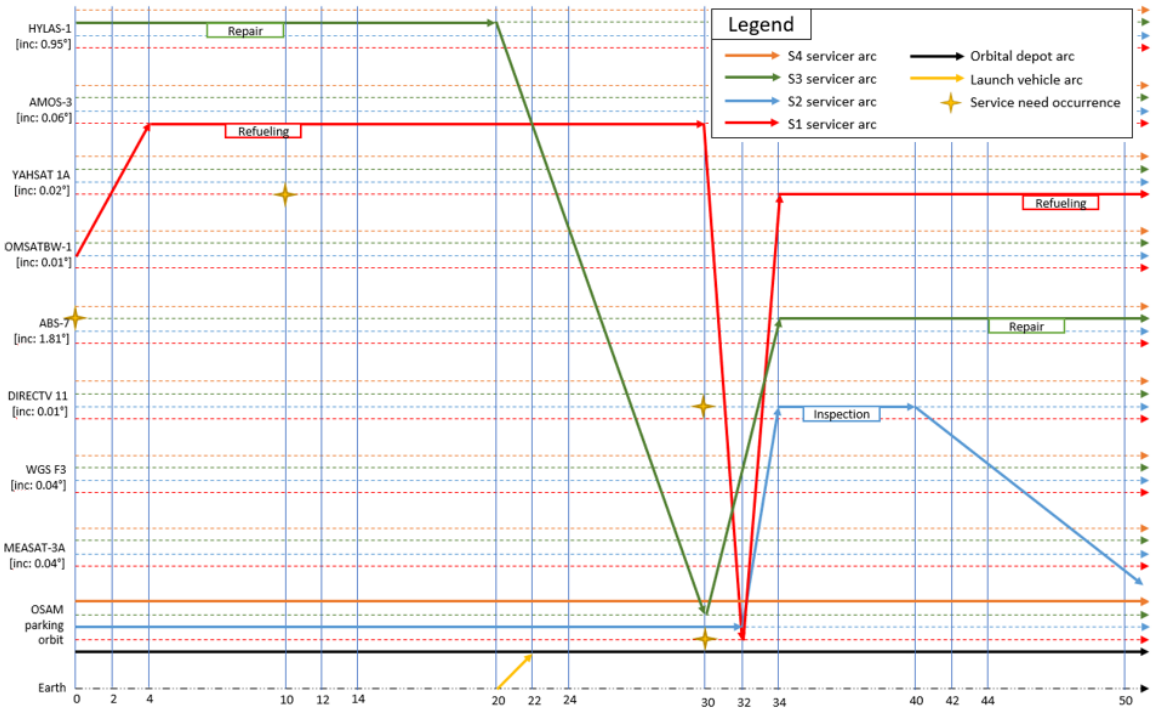

Figure 5. Breakdown of the operations of the four high-thrust specialized servicers over the initial 50 days of the planning horizon.

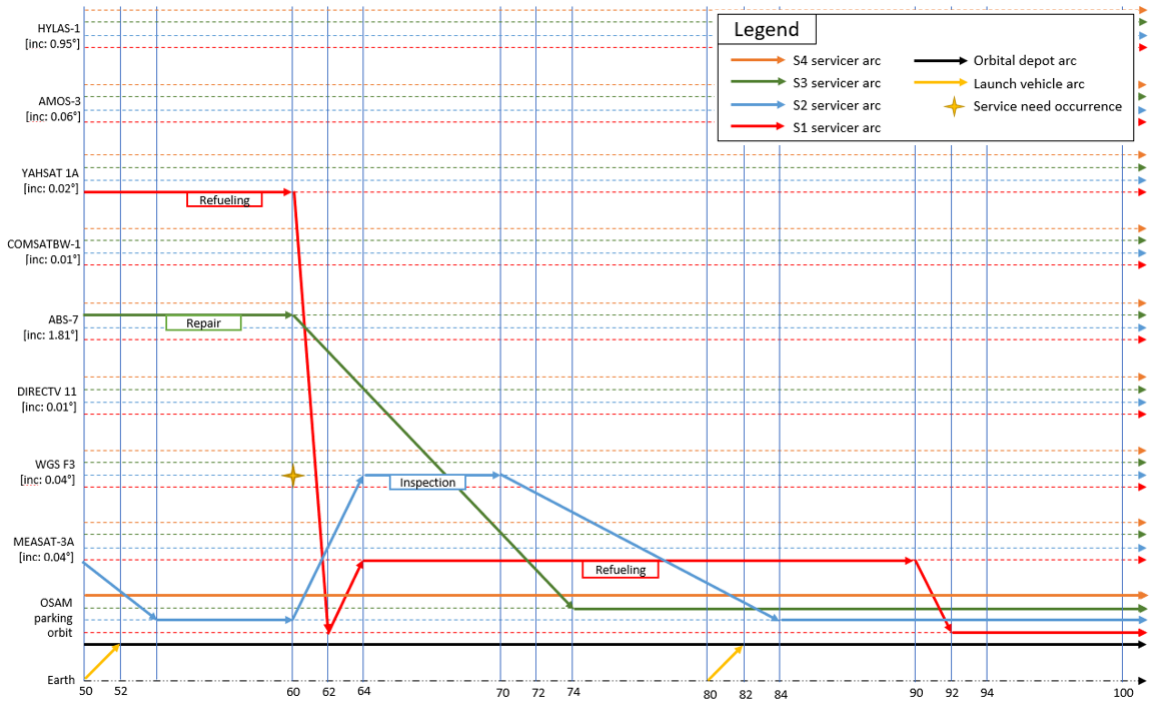

Figure 6. Breakdown of the operations of the four high-thrust specialized servicers over the final 50 days of the planning horizon.

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 marketing 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 (4 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. Figures 7-10 illustrate the value (i.e., initial investments subtracted from profits) of the monolithic and distributed infrastructures for four different 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 2 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 5-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.

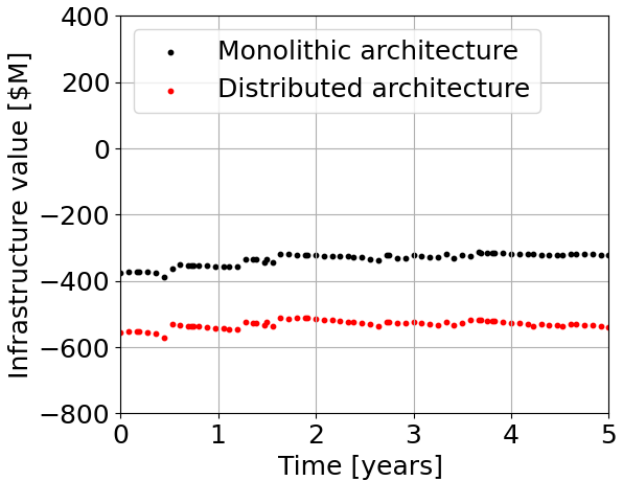

Figure 7. Values of the high-thrust monolithic and distributed architectures for a market condition of 30 customer satellites.

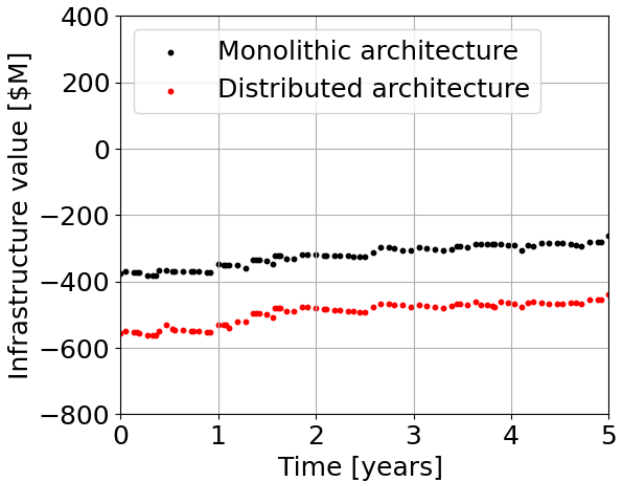

Figure 8. Values of the high-thrust monolithic and distributed architectures for a market condition of 57 customer satellites.

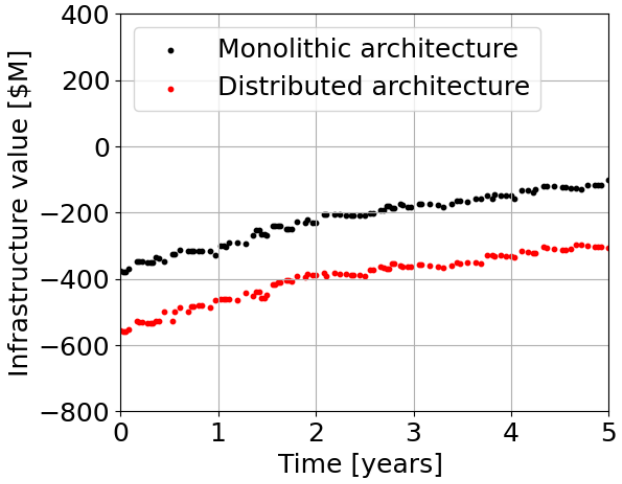

Figure 9. Values of the high-thrust monolithic and distributed architectures for a market condition of 114 customer satellites.

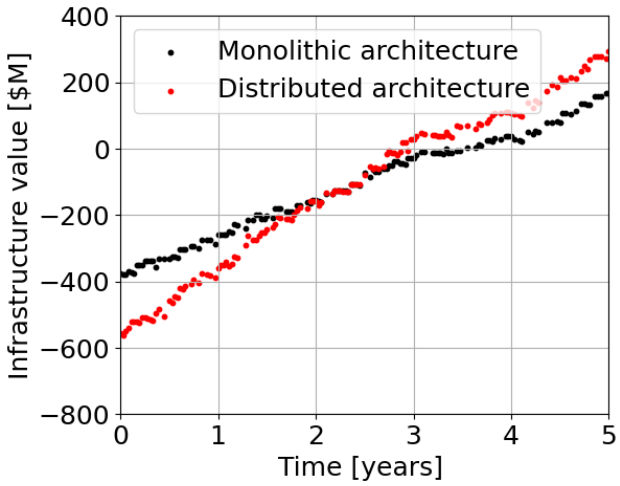

Figure 10. Values of the high-thrust monolithic and distributed architectures for a market condition of 228 customer satellites.

CONCLUSION

This paper generalizes the state-of-the-art network-based OOS logistics method to the multi-orbit case. 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 high-thrust and low-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.

Although the current planning tool can also simulate low-thrust and multimodal servicers, the simulations remain too long to run on a typical desktop computer due to the time taken by the low-thrust trajectory routine to run. Future work will include running the simulations on high-end computing clusters and devising a computationally efficient method to include the low-thrust trajectory model within the multi-orbit OOS framework. A promising research direction is to develop surrogate models of the low-thrust trajectory model using Gaussian Processes.

ACKNOWLEDGMENTS

This work is supported by the Defense Advanced Research Project Agency Young Faculty Award D19AP00127. The content of this paper does not necessarily reflect the position or the policy of the U.S. Government, and no official endorsement should be inferred. This paper has been approved for public release; distribution is unlimited.

APPENDIX: SERVICER TOOLS

Table A1. Notional service-tool mapping (represented by an 'X' in the table); from Reference 18.

	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	

REFERENCES

[1] J. Rainbow, "Space Tugs as a Service: In-orbit service providers are bracing for consolidation," *SpaceNews*, July 6, 2021. https://spacenews.com/space-tugs-as-a-service-in-orbit-service-providers-are-bracing-for-consolidation/ [retrieved 7/26/2021].

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

[3] S. Erwin, "Astroscale moving into GEO satellite servicing market," *SpaceNews*, June 3, 2020. https://spacenews.com/astroscale-moving-into-geo-satellite-servicing-market/ [retrieved 7/26/2021].

[4] "STP-1 (Space Test Program-1) / Orbital Express," *eoPortal Directory*. https://earth.esa.int/web/eoportal/satellite-missions/s/stp-1 [retrieved 7/26/2021].

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

[6] S. Zhao, P. Gurfil, and J. Zhang, "Optimal Servicing of Geostationary Satellites Based on Low-Thrust Transfers Considering Perturbations," *Acta Astronautica*, Vol. 39, No. 10, 2016, pp. 2219-2231. https://doi.org/10.2514/1.G001424

[7] C. Han, S. Zhang, and X. Wang, "On-Orbit Servicing of Geosynchronous Satellites Based on Low-Thrust Transfers Considering Perturbations," *Acta Astronautica*, Vol. 159, June 2019, pp. 658-675 https://doi.org/10.1016/j.actaastro.2019.01.041

[8] A. W. Verstraete, D. Anderson, N. M. St. Louis, and J. Hudson, "Geosynchronous Earth Orbit Robotic Servicer Mission Design," Journal of Spacecraft and Rockets, Vol. 55, No. 6, 2018, pp. 1444-1452. https://doi.org/10.2514/1.A33945

[9] 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, June 2013, pp. 107-126. https://doi.org/10.1016/j.actaastro.2013.02.005

[10] 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, March 2017, pp. 192-203. https://doi.org/10.1016/j.actaastro.2016.12.021

[11] K. Ho, H. Wang, P. A. DeTrempe, 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, pp. 1129-1138. https://doi.org/10.2514/1.A34663

[12] P. J. Sears, and K. Ho, "Impact Evaluation of In-Space Additive Manufacturing and Recycling Technologies for On-Orbit Servicing," *Journal of Spacecraft and Rockets*, Vol. 55, No. 6, 2018, pp. 1498-1508. https://doi.org/10.2514/1.A34135

[13] T. Sarton du Jonchay, "Modeling and Simulation of Permanent On-Orbit Servicing Infrastructures Dedicated to Modularized Earth-Orbiting Platforms," Master's Thesis, Univ. of Illinois at Urbana-Champaign, Champaign, IL, 2017.

[14] 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, Dec. 2018, pp. 109-121. https://doi.org/10.1016/j.actaastro.2018.10.040

[15] 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, 2006, pp. 485-498. https://doi.org/10.1016/j.actaastro.2005.12.014

[16] 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, 2020, pp. 844-850. https://doi.org/10.2514/1.A34701

[17] 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. https://doi.org/10.2514/1.A34978

[18] 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* (accepted).

[19] 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, 2020, pp. 446-463. https://doi.org/10.2514/1.A34628

[20] 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, 2018, pp. 365–381. https://doi.org/10.2514/1.A33905

[21] T. Ishimatsu, O. L. de Weck, J. A. 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, 2016, pp. 25–38. https://doi.org/10.2514/1.A33235

[22] K. Ho, O. L. de Weck, J. A. Hoffman, and R. Shishko, "Dynamic modeling and optimization for space logistics using time-expanded networks," *Acta Astronautica*, Vol. 105, No. 2, 2014, pp. 428–443. https://doi.org/10.1016/j.actaastro.2014.10.026

[23] K. Ho, O. L. de Weck, J. A. Hoffman, and R. Shishko, "Campaign-level dynamic network modelling for spaceflight logistics for the flexible path concept," *Acta Astronaut*ica, Vol. 123, 2016, pp. 51–61. https://doi.org/10.1016/j.actaastro.2016.03.006

[24] 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, 2019, pp. 12–20. https://doi.org/10.2514/1.A34168

[25] 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, 2020, pp. 80-92. https://doi.org/10.1016/j.actaastro.2020.01.031

[26] 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, Oct. 2009, pp. 303-315. http://dx.doi.org/10.1287/opre.1090.0721

[27] 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.

[28] 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. https://doi.org/10.1007/s10107-004-0559-y

[29] D. Izzo, "Revisiting Lambert's problem," *Celest. Mech. Dyn. Astron.*, vol. 121, no. 1, 2014. 10.1007/s10569-014-9587-y

[30] J. A. Sims, and N. Flanagan, "Preliminary design of low-thrust interplanetary missions," *AAS Astrodynamics Specialists Conference*, 1999.

[31] NORAD Two-Line Element Sets Current Data, CelesTrak. https://www.celestrak.com/NORAD/elements/ [retrieved 7/28/2021].

[32] B. R. Sullivan, and D. L. Akin, "Satellite Servicing Opportunities in Geosynchronous Orbit," *AIAA SPACE 2012 Conference and Exposition*, AIAA Paper 2012-5261, 2012. https://doi.org/10.2514/6.2012-5261

[33] Premiere for Europe: Jules Verne Refuels the ISS. The European Space Agency. https://www.esa.int/Science_Exploration/Human_and_Robotic_Exploration/ATV/Premiere_for_Europe_Jules_Verne_refuels_the_ISS [Retrieved 7/28/2021].

[34] C. Quilty, and A. Moeller, "January 2018 Satellite & Space Monthly Review", Quilty Analytics, Feb. 2018. https://www.quiltyanalytics.com/wp-content/uploads/2018_01-Satellite-Monthly-1.pdf [Retrieved 7/28/2020].