CEAS-GNC-2024-011

EuroGNC Navigation Guidance Control CEAS Bristol, UK June 11th-13th 2024

On-Orbit Servicing Strategies for Large Constellations via Hybrid Dynamic Programming

Ignacio Betti Lhuillier Student, Cranfield University, School of Aerospace Transport and Manufacturing, MK43 0AL, Cranfield, United Kingdom. nachobettil@gmail.com

Andrea Bellome Visiting Researcher, Cranfield University, Centre for Autonomous and Cyberphysical Systems, MK43 0AL, Cranfield, United Kingdom. Andrea.Bellome@cranfield.ac.uk

Leonard Felicetti Senior Lecturer in Space Robotics and GNC, Cranfield University, Centre for Autonomous and Cyberphysical Systems, MK43 0AL, Cranfield, United Kingdom. leonard.felicetti@cranfield.ac.uk

ABSTRACT

The increasing number of mega-constellations in LEO is not only increasing the number of satellites in orbit but potentially might trigger an increment in the amount of space debris. On-orbit servicing is seen as a viable solution to minimize the number of launches needed to maintain these constellations and keep the debris levels low. This paper focuses on trajectory planning for On-Orbit Servicing (OOS) by presenting a novel methodology based on dynamic programming hybridized with deterministic tree search strategies. This is used to schedule the best path for spacecraft to visit satellites that need servicing (such as repair, refueling or repositioning). This methodology is then used to conduct Monte Carlo simulations to study how the main characteristics of the spacecraft affect the performance of the servicing operation. The presented study aims to find the optimal number of servicers needed, the ΔV required to perform the operations, and the time of flight between satellites.

Keywords: Dynamic Programming; Beam Search; Multi-target Mission Analysis; Monte Carlo; Tree Exploration

Nomenclature

a	=	Semi-major axis
b_w	=	Beam Width
DP	=	Dynamic Programming
e	=	Eccentricity
i	=	Inclination
M	=	Mean anomaly
N_{fails}	=	Total number of failed satellites in one simulation
$N_{rep,sc}$	=	Total repairs performed by a spacecraft in one simulation
$N_{rep,tot}$	=	Total number of repairs performed in one simulation
R_{tot}	=	Repaired satellites
$R_{tot,sc}$	=	Spacecraft relative performance
ToF	=	Time of Flight

$ToF_{max,s}$	=	Maximum ToF per satellite transfer
$ToF_{max,t}$	=	Maximum ToF per tour
t_{opt}	=	Optimal time
β	=	Weibull distribution shape parameter
$\Delta V_{max,s}$	=	Maximum ΔV per satellite transfer
$\Delta V_{max,t}$	=	Maximum ΔV per tour
λ	=	Weibull distribution scale parameter
μ_{SL}	=	Expected Satellite Life
ω	=	Argument of periapsis
Ω	=	Right-ascension of the ascending node
σ^2_{SL}	=	Variance of Satellite Life

1 Introduction

On-orbit servicing is becoming a viable solution to extend the life of current satellite assets by performing standardized operations aiming to refuel, repair, or recover existing satellites in orbit. It is predicted that the European on-orbit servicing market will reach $1 billion by 2030, as it is more cost-effective than launching a new satellite[1]. Future satellites could be designed with less fuel and larger instruments that could be assembled in orbit from modular components that are readily exchangeable [1]. Moreover, technological advances in space operations autonomy and robotics are ready to revolutionize the paradigm of satellite maintenance [2]. Within the next 5 to 10 years, routine spacecraft refueling, and hardware upgrades could become a reality, eliminating the need to rely on satellites with decades-old hardware and technology. This would significantly reduce the cost of space operations and enable new and more ambitious missions [3]. These trends are likely to continue in the coming years, making OOS an increasingly important tool for maintaining and operating satellite constellations.

Large constellations are boosting the number of satellites in the Low-Earth Orbit (LEO). It is therefore important to either de-orbit as quickly as possible any satellites that have reached the end of their operational life before they can pose a threat to other satellites [4, 5], or decrease the number of satellites deployed by keeping them on service. As a solution, on-orbit servicing (OOS) appears with the capabilities of extending satellite life through restoring, repairing, inspecting, or relocating (including de-orbiting when life extension is not viable) those inoperative satellites. Furthermore, the use of standardized components and platforms across a large constellation could make it possible to store spare parts in orbit, which could be used to repair satellites that have suffered a failure quickly. This results in a more cost-effective and time-efficient solution than launching a completely new satellite.

As OOS becomes more feasible and interesting for the client, diverse studies on how to encourage servicing appear. For example, Carvalho and Kingston provided metrics to quantify how servicers relate to clients and how they affect the different services provided [6]. The methodology used to analyse the relationship between the agents served as a baseline for this paper. However, their focus was set on the economic feasibility by analysing the net value of OOS. The investigation of Yao et al. moved forward presenting a method to simulate the life-cycle of the satellite within a constellation and a method for decision-making using dynamic programming for tree exploration [7]. This methodology contributed to the analysis; however, the focus was on maximizing the utility but not involving trajectory optimization. On the other hand, Borelli et al. performed an investigation about trajectory design for large constellations OOS [8] but with a main focus on close proximity operations.

However, OOS might involve multi-target tours of the servicers, which are required to optimize the sequence of the satellites to be serviced within the constellation. The novelty of this paper arises in the development of a tool to automatically schedule the best tour to service the satellite within a mega

[1] https://www.marketsandmarkets.com/Market-Reports/on-orbit-satellite-servicing-market-206789424.html

constellation using hybrid dynamic programming principles [9, 10], coupled with deterministic beam search [11], to reduce the computational effort. This scheduler determines how a servicer spacecraft interacts with a population of satellite clients. The paper also uses the scheduler to perform a Monte Carlo analysis to study the service's behaviour and characteristics to find an optimal architecture to provide it.

Given the inherent complexity, similar problems to the one presented in this paper are being analyzed recurrently in the Global Trajectory Optimization Competition (GTOC). GTOCs are international competitions first proposed by the Advanced Concept Team (ACT) at ESA in 2005 [12]. Especially related to this publication was the GTOC 9th edition, where the problem definition was Multi-mission removal of sun-synchronous debris. Genetic algorithms are one of the most used approaches in GTOC as applied in [13] [14]. However, such approaches are stochastic in nature and thus do not guarantee the convergence to optimal solutions. Moreover, the performances of such solvers usually depend upon a number of user-defined parameters that need to be properly tuned, and such tuning is problem-specific. These problems are highly mitigated using dynamic programming [15] to guarantee global optimality within limited computational effort.

To achieve this, section 2 first defines the mathematical model and the problem formulation, with simplified orbital dynamics, as well as the servicer orbital transfers. Then, section 3 defines the satellites failure model based on Weibull distribution (subsection 3.1) and the tree exploration to determine the best tour to visit the satellites via hybrid dynamic programming (subsection 3.2). The term hybrid refers to the use of beam search [11] in cases where the number of solutions grows larger than the computer resources can handle. Section section 4 presents the results obtained under the analysis of four different scenarios to find the optimal mission architecture via Monte Carlo analysis. The concluding section, section 5, draws the final remarks obtained from this analysis and outlines the future developments in this line of research by the authors.

2 Problem Formulation

OOS might require to visit multiple satellites with one single platform to refuel or repair them. The trajectory design of OOS missions thus consists in a mixed-integer non-linear programming problem (MINLP) [16] [17]. In MINLP, the combinatorial problem of selecting the visiting order of satellites, encoded in vector X via their IDs, is coupled with optimal control theory, whose variables, i.e., the satellites visiting epochs, are encoded in vector y. Thus, a general MINLP has the following structure:

$$\begin{aligned} \text{Minimize: } & f(X, y) \\ \text{Subject to: } & g_i(X, y) \geq 0, \forall i = 1, ..., m_{in} \\ & X_{lb} \leq X \leq X_{ub} \\ & y_{lb} \leq y \leq y_{ub} \end{aligned} \tag{1}$$

where: $f(X, y)$ is the cost function, that, in this paper, encodes a positive real number defined by the total impulse by mass unit needed to visit satellites needing OOS, i.e, the ΔV. In addition, (X, y) represents a chain of unique satellites to visit, defined by an ID and the epoch when they are visited. $g_i(X, y)$ represent the inequality constraints of the problem at hand. In the OOS case under concern, these are the mission duration, the transfers durations, the corresponding mission ΔVs and transfers ΔVs. Please note that m_{in} is the number of inequality constraints; (X_{lb}, y_{lb}) and (X_{ub}, y_{ub}) represent box constraints, i.e., lower and upper bounds for (X, y), respectively, as per Table 1 in the following results section 4.

A simplified dynamical model for satellites' dynamics is used in this paper, taking into account Earth's oblateness as the primary source of orbital perturbation, i.e., the harmonic J_2 of the Earth. Such a model assumes that orbital perturbations due to Earth's oblateness are only considered in its secular effects on satellites' right-ascension of the ascending node (Ω), the argument of periapsis (ω) and mean anomaly (M), while semi-major axis (a), eccentricity (e) and inclination (i) remain constant. Time variations of Ω, ω and M are thus given by:

$$\dot{\Omega} = \frac{d\Omega}{dt} = -\frac{3}{2}\sqrt{\frac{\mu}{a^3}}\frac{J_2\cos(i)}{\left(1-e^2\right)^2}\left(\frac{r_E}{a}\right)^2 \tag{2}$$

$$\dot{\omega} = \frac{d\omega}{dt} = \frac{3}{4}\sqrt{\frac{\mu}{a^3}}\frac{J_2\left(5\cos(i)^2-1\right)}{\left(1-e^2\right)^2}\left(\frac{r_E}{a}\right)^2 \tag{3}$$

$$\dot{M} = \frac{dM}{dt} = \sqrt{\frac{\mu}{a^3}} + \frac{3}{4}\sqrt{\frac{\mu}{a^3}}\frac{J_2\left(3\cos(i)^2-1\right)}{\left(1-e^2\right)^{3/2}}\left(\frac{r_E}{a}\right)^2 \tag{4}$$

where μ and r_E are the Earth's gravitational parameter and equatorial radius, respectively. Such a model has been used extensively in literature [18] and shows good accordance with propagators such as SGP4 [19, 20]. From Eqs. (2) to (4), the values for Ω, ω and M at time t are then given by:

$$\Omega(t) = \Omega_0 + \dot{\Omega}(t-t_0) \tag{5}$$

$$\omega(t) = \omega_0 + \dot{\omega}(t-t_0) \tag{6}$$

$$M(t) = M_0 + \dot{M}(t-t_0) \tag{7}$$

where Ω_0, ω_0 and M_0 are the values at a reference time t_0. In this way, one has satellite states at any time t.

To model the servicing-spacecraft transfers between two satellites, the approximate ΔV computation described by Shen and Casalino [20] is employed. In such a model, the spacecraft is assumed in rendezvous conditions with constellation satellites when it matches both position and velocity vectors. The cost to connect different satellites is thus assumed to be equivalent to the ΔV required to change a, e, i and Ω between the two satellites' orbits at different times t_0 and t. Two cases need to be considered:

1) the transfer time $t - t_0$ is such that the satellites' Ω can be aligned naturally due to the J_2 effect. In this case, one considers only the effect of changing a, e and i.
2) the transfer time $t - t_0$ is not enough for Ω natural alignment given by the J_2 drift. In this case, one also considers the Ω change.

The interested reader is referred to [20] for further details, and a summary of the procedure is also presented in Appendix. The relative error between the approximation presented and exact solutions with multi-impulses trajectories [21] are between 5% and 6% maximum [20]. This is deemed sufficient to truthfully represent the search space of transfer options for the preliminary design and analysis presented in this paper. With this approximation, the ΔV cost connecting two objects in vector X only depends upon the epochs t at which they are visited, encoded in vector y.

3 Satellite Failure Models and Tree Exploration Methods

OOS in mega-constellation is a multi-target mission design problem where a servicer needs to visit failed satellites within the mega-constellation. To solve this problem, the scheduler needs to find the optimal sequence of failed satellites that fixes the most with minimal ΔV usage. This requires constructing chains of potential satellites to be visited and then selecting the optimal one.

The proposed scheduling methodology is based on the architecture presented in Figure 1. The Figure illustrates a loop that is repeated every day starting from a reference epoch. In particular, subsection 3.1 describes how the simulation sets some of the satellites as non-operative; then, subsection 3.2 shows how to schedule the servicers with the best path to repair them. The scheduler tries to define an optimal tour for the spacecraft to service the maximum number of satellites that are set as non-operative. If a feasible tour is found, the fixed satellites become operative again and the loop is finished.

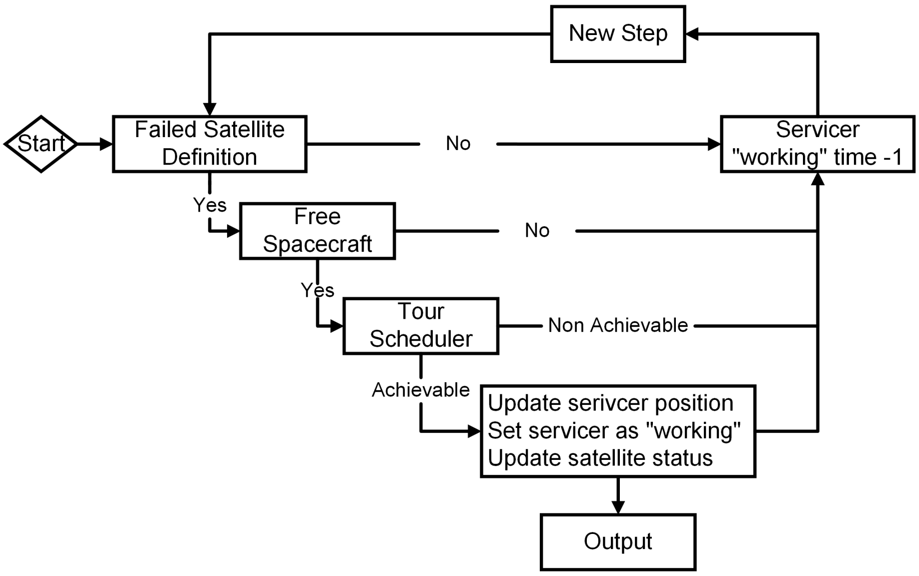


Fig. 1 Simulation iteration loop

3.1 Satellite failure using Weibull Distribution

Satellites are considered out of service when they can no longer operate normally. This can be due to a variety of causes, such as fuel depletion, electronic failure, or debris impact. While the algorithm used to model the satellite status can handle any type of failure, this paper is tuned to model more accurately failures due to wear-out, not considering infant mortality. The satellites failure is modelled using the Weibull distribution, commonly adopted to model failures that are consequence of continued use [22].

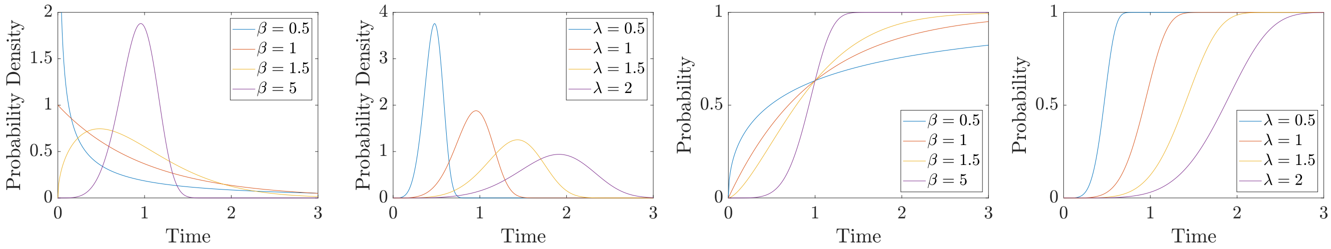
The Weibull probability density function (pdf) is given by [23]:

$$pdf(t,\lambda,\beta) = \begin{cases} \frac{\beta}{\lambda}\left(\frac{t}{\lambda}\right)^{\beta-1} e^{-(t/\lambda)^{\beta}}, & t \geq 0 \\ 0, & t < 0 \end{cases} \tag{8}$$

and cumulative distribution function (cdf) is given by:

$$cdf(t,\lambda,\beta) = \begin{cases} 1 - e^{-(t/\lambda)^{\beta}}, & t \geq 0 \\ 0, & t < 0 \end{cases} \tag{9}$$

where: t is the satellite lifetime, β is the shape parameter, and λ is the scale parameter. Figure 2 shows the pdf and cdf for different values of β and λ. From Figure 2a, strictly decreasing pdf functions are obtained with $\beta \leq 1$, that are good to model infant mortality, while with $\beta > 1$, a peak on pdf curves is found, proper to model wear-out failure cases. The case where $\beta = 1$ leads to the exponential distribution that models cases where hazard rates are independent of time as debris impact [22]. On the other hand, λ defines the position of the peak with respect to t, as well as the spread of the probability, seen in the width of the peak Figure 2b. On the other hand, Figure 2c and Figure 2d show the evolution of the cdf with respect to t for different values of β and λ, respectively, corresponding to the pdf shown in Figure 2a and Figure 2b.



(a) Effect of β in the Weibull density function ($\lambda = 1$)

(b) Effect of λ in the Weibull density function ($\beta = 5$)

(c) Effect of β in the Weibull cumulative function ($\lambda = 1$)

(d) Effect of λ in the Weibull cumulative function ($\beta = 5$)

Fig. 2 Effect of β and λ on the density and cumulative Weibull distribution functions.

The test case scenarios under consideration in this paper consider values of β and λ set in such way that the expected life for the satellite is 7.5 $years$ (μ_{SL}) with a variance of 3.5 $years^2$ (σ^2_{SL}). These values are based on the analysis of the current lifetime of general satellites whose design life-cycle was initially planned to be 5 years[2].

3.2 Tree construction using Hybrid Dynamic Programming

Once some of the satellites within the constellation are set as non-operative, an optimal OOS is defined such that it maximizes the repairs performed while minimizing the ΔV required for the service. The MINLP problem in Eq. (1) thus needs to be solved, with $f(X, y)$ equal to the total ΔV needed by one spacecraft to visit the satellites. However, analysing in detail each servicing tour is unfeasible for large constellations, due to the number of combinations to be evaluated (e.g., 10 satellites correspond to 3.6 million combinations). Thus, one transcribes the MINLP from Eq. (1) into a discrete combinatorial problem by making the visiting epochs t to vary discretely on grids. In this way, one can organize the space of possible solutions, i.e., the tours, in a tree-graph [24]. Such transcriptions allows dynamic programming principles to be applicable, enabling optimal solutions to be captured in the transcribed space with limited computational effort [10]. The problem is discrete because satellites are represented by integers and the visiting time is discretized using a grid, and it is combinatorial because it analyses the different paths (combinations) that emerge from those nodes.

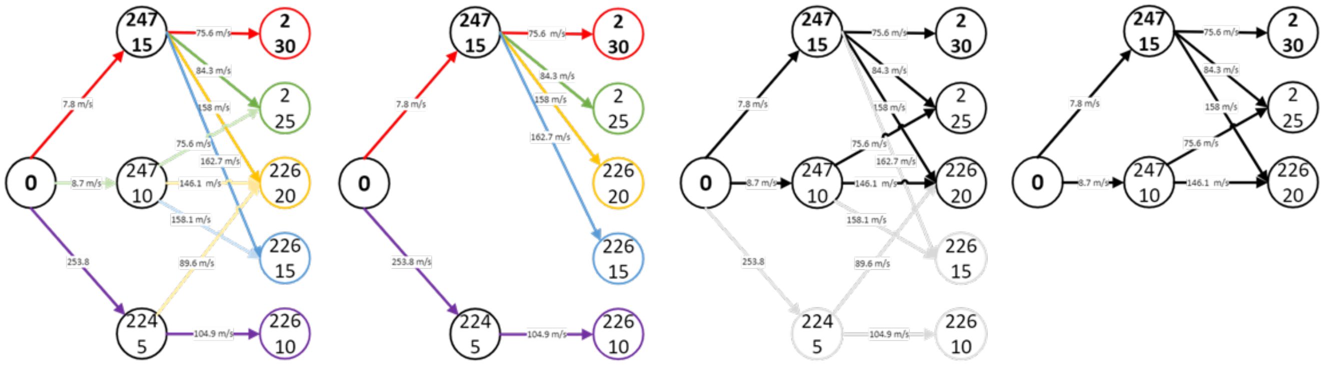
The transcribed tree-graph is formed by nodes and edges [24]. Each node is a potential repair characterized by the satellite repaired (through its ID), and the time when it is repaired. Each edge connecting two nodes represents the cost of performing that transfer, i.e., the ΔV. Finally, a tour is defined as a branch that begins in the initial node and ends when there is not any other node to connect. When expanding the tree of possible tours, the following pruning criteria are adopted (see also section 4):

- Two nodes can not be connected if the ΔV between them exceeds a given threshold.
- A tour can not be further expanded if the accumulated ΔV exceeds a given threshold.
- A tour can not be further expanded if the accumulated Time of Flight (ToF) exceeds a given threshold.

To efficiently explore the tree of potential solutions, dynamic programming techniques are adopted [9, 10, 15]. These allow to agilely explore the search space of possible nodes, guaranteeing optimal solutions. However, when the number of nodes to be explored is too high to be handled by a standard laptop (e.g., CPU of 3.5GHz with 16GB of RAM), an issue known as the curse of dimensionality [9] arises. This is mitigated via beam search strategy [11], where only a fixed number of nodes are kept for

[2] https://aerospace.org/story/majority-satellites-exceed-design-life

further expansion. The resulting algorithm is thus a hybrid dynamic programming approach. Figure 3 represents such an approach.



(a) Exploration tree before (left) and after (right) applying the dynamic programming procedure

(b) Exploration tree before (left) and after (right) applying the beam search procedure

Fig. 3 Tree construction and pruning using hybrid dynamic programming, Figure 3a shows how the tree is constructed using dynamic programming, while Figure 3b shows how the tree is pruned with the application of beam search.

All branches are expanded from a given node discarding the ones that require more ΔV than the allowed. Dynamic programming is triggered when multiple tours arrive at a common node, meaning that both paths have the same future branches; however, one path has necessarily consumed less ΔV to arrive at the actual node. Therefore, continuing both branches is unnecessary, as this would over-stress computer resources. In this case, only the branch with the lowest ΔV consumption is continued (Figure 3a). This corresponds to a single-objective dynamic programming optimization [10].

Beam search then reduces the computational effort by pruning the less promising branches in the pool of tours. This is triggered only when the number of tours to be kept in memory for further expansion is bigger than a pre-defined maximum value. Such value is known as the beam width (b_w). In this case, the tours are sorted by ΔV and only the best b_w are kept for further consideration. Figure 3b shows an example of the application on beam search, with $b_w = 5$.

4 Results

The developed scheduler is used to analyse the viability and the architecture needed to service the OneWeb constellation. This constellation is formed by 616 satellites in the used database update (April 2023). For the analysis, two architectures are analysed:

- A servicing performed by a single spacecraft (subsection 4.1).
- A servicing performed by multiple spacecraft (subsection 4.2).

Monte Carlo analysis is performed with 500 simulations. Each simulation modelled service delivery over 1500 days, to analyse the trend of ΔV used, repairs made and saturation of the servicers. A randomized satellite failure process, detailed in subsection 3.1, is implemented within each simulation. This process randomly designates some satellites as non-operative on each day within the simulated timeframe. In this paper, each set of inputs that define a Monte Carlo simulation is called Scenario. Four scenarios are studied overall: one for a single servicer and three for multi-servicers. A summary of the constraints characterising each scenario is presented in Table 1.

Each scenario is characterized by the maximum ΔV and ToF that the spacecraft can use to transfer from one satellite to the other ($\Delta V_{max,s}$ and $ToF_{max,s}$, respectively), the maximum ΔV and ToF that limit a tour length and the number and distribution of the spacecraft along the constellation ($\Delta V_{max,t}$ and

$ToF_{max,t}$ respectively). More information about the selection of the properties is detailed in subsection 4.1 for Scenario 0 and subsection 4.2 for Scenarios 1, 2 and 3.

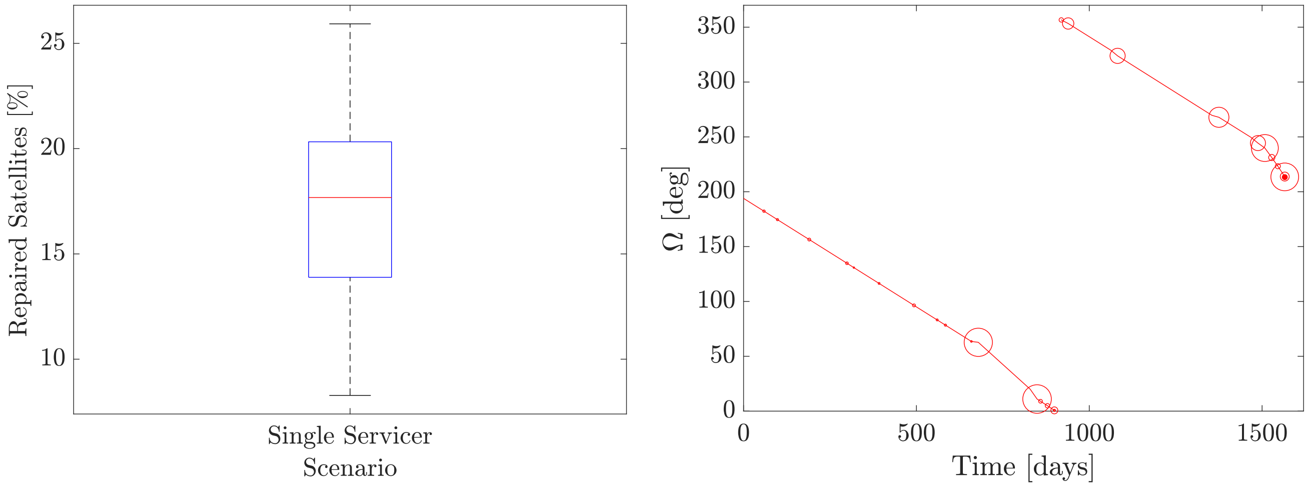
Table 1 Simulated scenarios properties definition

	Single Servicer	Multiple Servicer		
Parameters	Scenario 0	Scenario 1	Scenario 2	Scenario 3
$\Delta V_{max,s}$ [m/s]	400	400	400	Variable (Table 3)
$\Delta V_{max,s}$ [m/s]	1200	1200	1200	1200
$ToF_{max,s}$ [days]	20	10	20	20
ToF step [days]	2.5	1.25	2.5	2.5
$ToF_{max,t}$ [days]	100	100	100	100
Number of servicers	1	13 (Table 2)		14 (Table 3)

4.1 Single Spacecraft Service

To replicate the behaviour of the single servicer study case, a Monte Carlo analysis is run following Scenario 0 presented in Table 1.

$\Delta V_{max,s}$ is set to 400 m/s allowing Ω changes up to 2.5°. As for $\Delta V_{max,t}$, a value of 1200 m/s is deemed appropriate as a threshold to limit the ΔV used by the servicers. $ToF_{max,s}$ was selected based on a trial-and-error process. This process involved simulating scenarios with $ToF_{max,s}$ from 3 days to 40 days. As a result, values lower than 10 days highly increased the ΔV needed without significantly decreasing repairing time or the repairs performed. While values higher than 20 days lower the repairs performed by the servicers.



(a) **(b)**

Fig. 4 Single servicer results. Figure 4a shows using a box plot the percentage of repairs performed along every simulation, while Figure 4b shows an example of the repairs performed by one simulation. The circles' size can be used to compare the ΔV used in each transfer.

Figure 4a summarizes the results of the simulations. In particular, it shows how the quartiles of the distribution of the percentage of repaired satellites (Equation 10) are distributed over a set of 500 Monte Carlo simulations using a box plot. The percentage of repaired satellites (R_{tot}) is used as a metric of the efficiency of the service and it reads as:

$$R_{tot}\ [\%] = 100 \cdot \frac{N_{rep,tot}}{N_{fails}} \tag{10}$$

where $N_{rep,tot}$ is the total number of repairs performed in one simulation and N_{fails} is the number of satellites that failed in the same simulation.

In Figure 4a, the upper and lower lines represent the maximum and the minimum values of the percentage of repairs obtained. The lower and the upper sides of the box represent the first quartile (where 25% of the simulations fall) and the third quartile (where 75% of the simulations fall), respectively. The red line represents the second quartile or median of the distribution. Thus, it can be deduced that a single spacecraft can provide service to a maximum of 26% of the potential satellites needing servicing. It is worth adding that in this specific case, the average repairs in the distribution is 17.4%. In addition, the limited $\Delta V_{max,s}$ capabilities imposed in this scenario prevent the servicer from moving across different orbital planes of the constellation. This is represented in Figure 4b, where the evolution of Ω over time is shown for one simulation of the Monte Carlo analysis. The near-linear behavior of plots is mainly due to the J2 effects, demonstrating that the servicer can only change within a small set of near orbital planes. If servicing the entire constellation is desired, then either multiple spacecraft or larger $\Delta V_{max,s}$ are needed.

4.2 Multiple Spacecraft Service

To increase the number of repairs, three multiple spacecraft cases were analysed by setting:

- the $ToF_{max,s}$ to 10 days.
- the $ToF_{max,s}$ to 20 days.
- the $\Delta V_{max,s}$ of each spacecraft to a manually tuned value to improve the repairs.

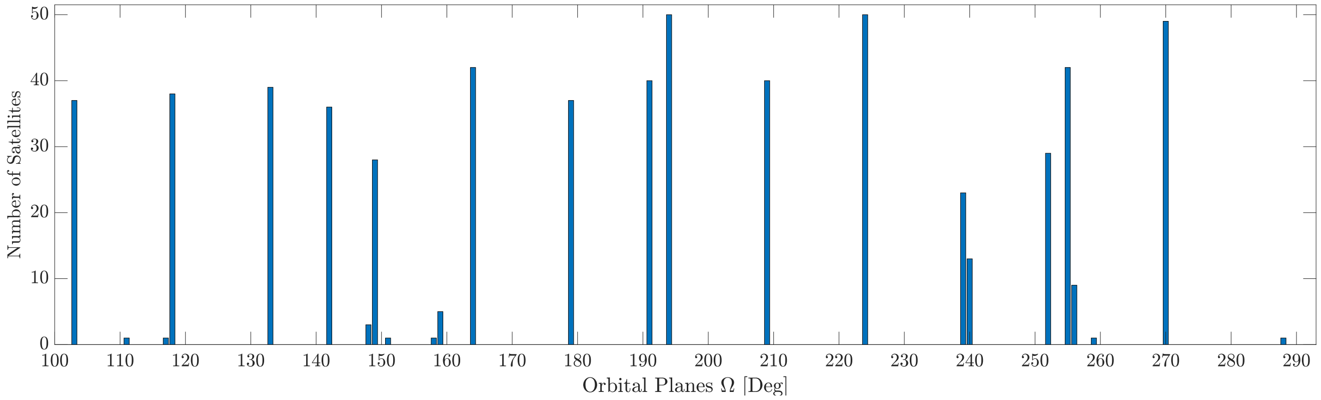


Fig. 5 OneWeb's satellites distribution over Ω

The OneWeb constellation is architecturally organized into groups of planes in close proximity as shown in Figure 5. These groups are then separated by larger distances in terms of $\Delta\Omega$. With the defined $\Delta V_{max,s}$, i.e., 400 m/s, one spacecraft can service Ω bands of maximum 4°. This division allows for sorting the population of OneWeb constellation in 13 bands, with a single servicing spacecraft assigned to each of them, as described in Table 2.

Table 2 Multiple servicers: Bands configuration

Band	Spacecraft Ω_0 [Deg]	Assigned Satellites	Band	Spacecraft Ω_0 [Deg]	Assigned Satellites
1	103	37	8	194	90
2	118	39	9	209	40
3	133	39	10	224	50
4	142	36	11	239	36
5	149	32	12	255	81
6	164	42	13	270	49
7	179	37			

To assess the impact of $ToF_{max,s}$ on the service's behaviour, the first two scenarios are compared. A third scenario is also proposed to improve Scenario 2, with the $\Delta V_{max,s}$ assigned to each spacecraft individually tuned to increase the number of repairs. This is because each spacecraft services an area with distinct properties, such as the number of satellites and the distance between them. The properties defining Scenarios 1, 2, and 3 are summarized in Table 1 (Scenario 1, 2 and 3). Focusing on Scenarios 1 and 2, Figure 6a shows the results of the sensitivity analysis of ToF. This illustrates two promising architectures for achieving constellation coverage up to 98.54% of the satellites. As can be seen in Figure 6a in both cases, half of the simulations repair more than 87% of the potential repairs.

Due to the variations in the number and distribution of satellites in each serviced band, an analysis of their relative performance (Equation 11) is conducted. The following equation represents the spacecraft's relative performance (R_{sc}).

$$R_{sc}\ [\%] = 100 \cdot \frac{N_{rep,sc}}{N_{rep,tot}} \tag{11}$$

$N_{rep,sc}$ is the total repairs performed by a spacecraft in a simulation.

The results are shown in Figure 6b. The relative performance metric calculates the percentage of total repairs performed in each band, represented by the green and blue lines in the figure, and compares it to the percentage of expected repairs, represented by the orange bars in the plot. The percentage of expected repairs is calculated based on the number of satellites assigned to each of the bands compared to the total number of satellites in the constellation. Given that all satellites have the same probability of failure, each band's performance is expected to align with the expected repair percentage for each band in an optimal service scenario.

It is worth to note that Figure 6b shows that both configurations have similar results. However, bands 8 and 12 have a way lower percentage of repairs compared to the values that are potentially expected in this configuration. Band 13, on the other hand, outperformed the optimal configuration. The first of the problems arose because both bands 8 and 12 are constituted by two orbital planes equally crowded but not sufficiently closed to be reached by the same service spacecraft with the $\Delta V_{max,s}$ provided in Scenario 1 and 2. On the other hand, the option that the servicers were running out of time on their multi-target satellite serving tour was discarded because the occupancy of the servicers in both scenarios was always below 18%, as shown Figure 7a. The higher performance of band 13 suggests a potential invasion of other bands by the servicer assigned to that particular band.

Figure 7a and Figure 7b show the average spacecraft occupancy (defined as the percentage of the time spent in the servicing tours with respect to the simulation time) and the total ΔV_{total} used by the servicers in average during the entire simulation. It is worth noting that using a $ToF_{max,s}$ of 10 days

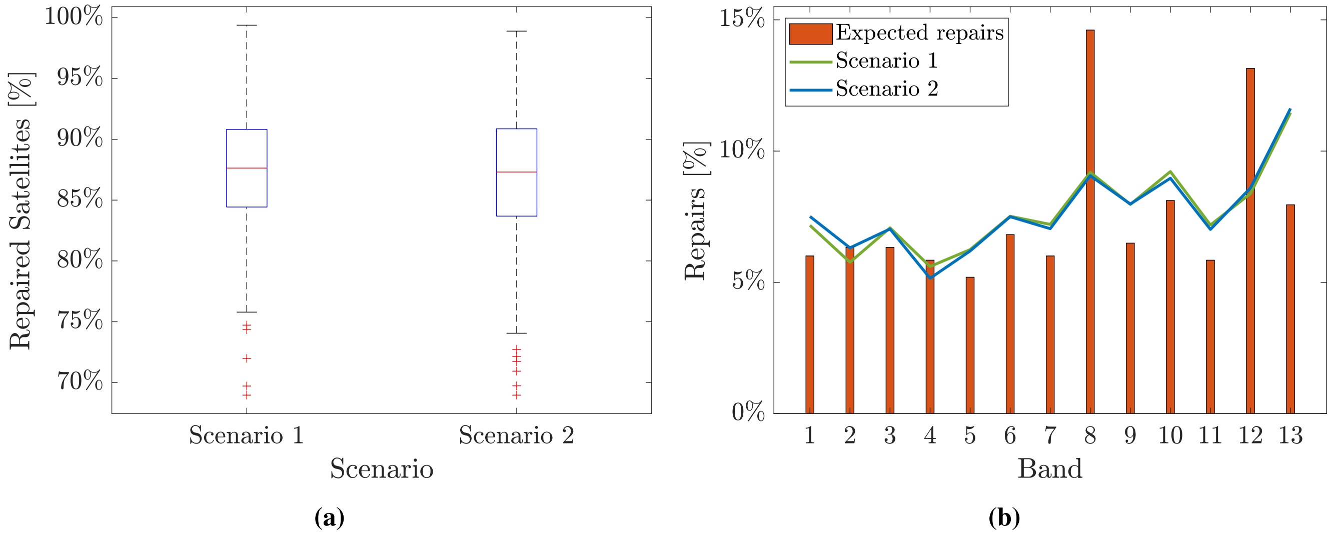


Fig. 6 Repaired satellite of Scenario 1 and Scenario 2 results. In Figure 6a the percentage of failed satellites repaired by the spacecraft and in Figure 6b the band relative performance compared to the assigned satellite per band.

lowers the occupancy of the servicers but increases the ΔV used. These results suggest that increasing the $ToF_{max,s}$ lowers the ΔV needed by the servicer.

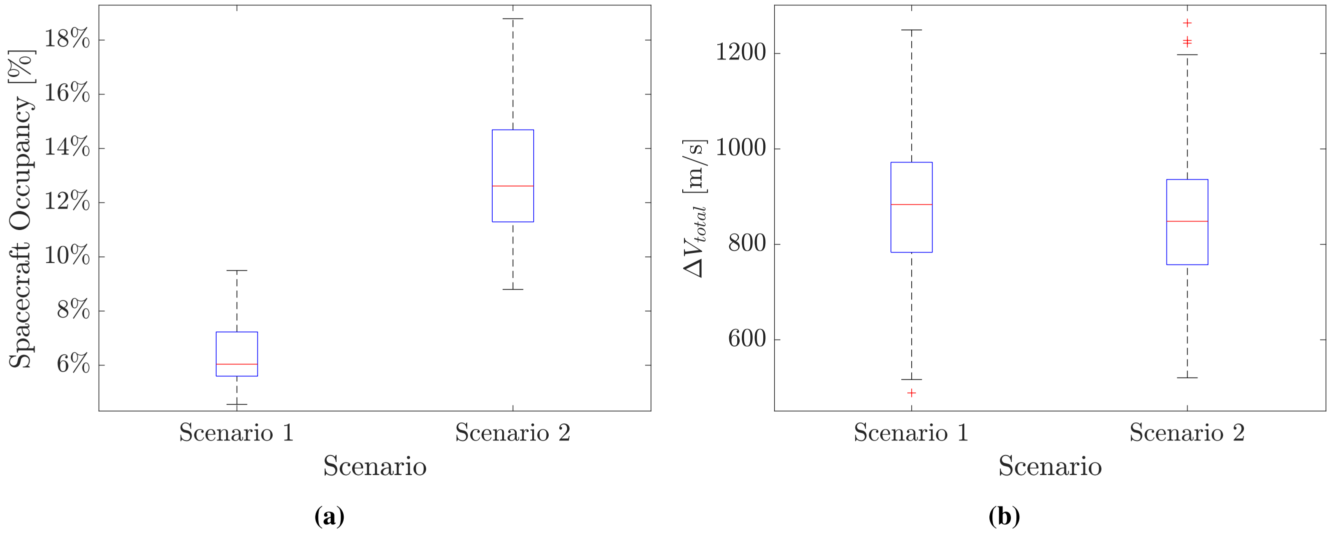
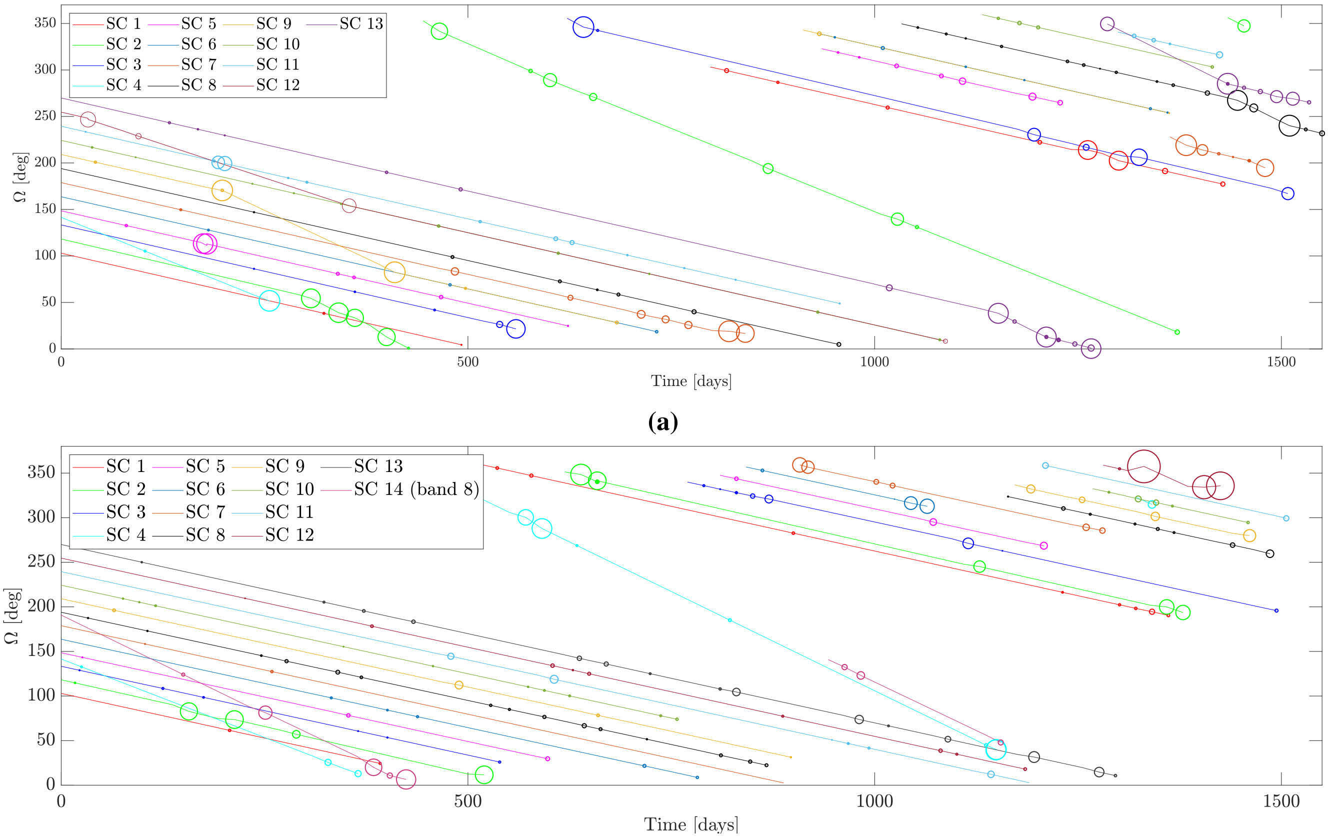


Fig. 7 Occupancy and total ΔV of Scenario 1 and Scenario 2 results. In Figure 7a the percentage of the simulation time that the spacecraft are servicing. In Figure 7b the ΔV that each spacecraft uses during the simulation time

Given that bands 8, 12 and 13 were performing far from expected and that each band has unique characteristics, a new scenario is developed to address these issues and improve the ΔV efficiency of all the spacecraft. Figure 8a shows the Ω of each of the spacecraft during one of the simulations, represented with coloured lines, and the instants when the repairs occur, with circles of size proportional to the ΔV used in the transfer. It is worth noticing not only that, in some cases, the servicers are invading other bands (e.g. spacecraft 9 at day 400 entering band 6) but that the planes of the constellation drift at different speeds, preventing the service from achieving their maximum potential: planes with different Ω drifts escape from the bands and can not be serviced until they enter another band.



(a)

(b)

Fig. 8 Ω of each spacecraft and performed repair over the simulation time for Scenario 2 (Figure 8b) and 3 (Figure 8a). Note that breaks in lines represent a servicer wrap from 0° to 360°

These considerations led to a search for improvements by tuning the $\Delta V_{max,s}$ of each of the spacecraft to match the optimal performance. After the iteration process, a new scenario (Scenario 3), based on Scenario 2, was proposed implementing two major changes:

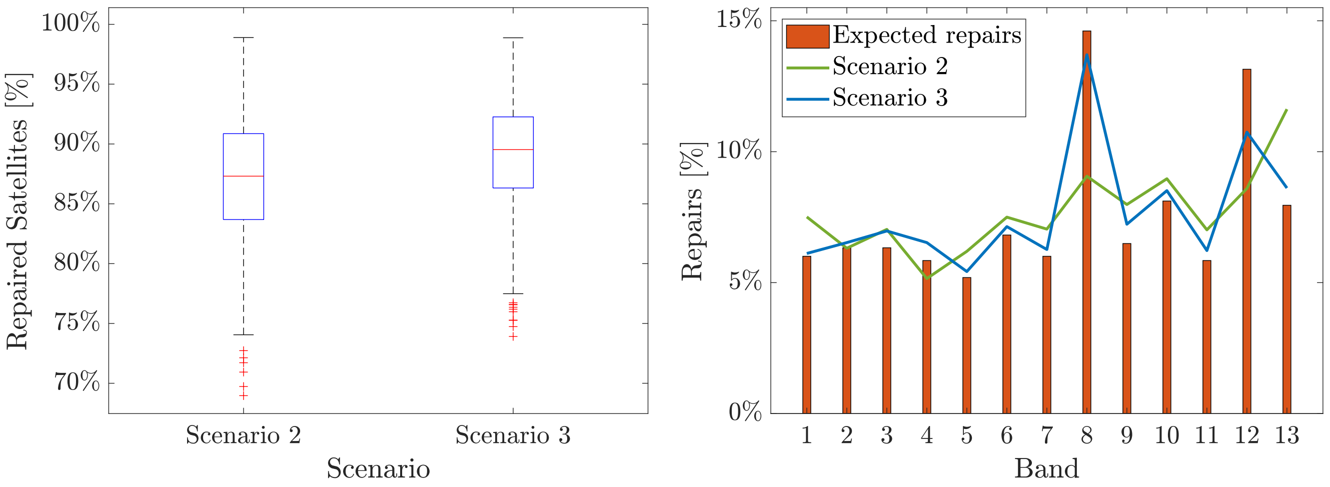
- Tuning of $\Delta V_{max,s}$ for each of the spacecraft.
- Addition of a second servicer for band 8

Table 3 Scenario 3 band configuration and servicers ΔV

Band	Spacecraft Ω_0 [Deg]	Assigned Satellites	$\Delta V_{max,s}$ [m/s]	Band	Spacecraft Ω_0 [Deg]	Assigned Satellites	$\Delta V_{max,s}$ [m/s]
1	103	37	400	8	191	40	400
2	118	39	200		194	50	400
3	133	39	200	9	209	40	200
4	142	36	400	10	224	50	200
5	149	32	200	11	239	36	200
6	164	42	200	12	255	81	1000
7	179	37	200	13	270	49	200

The inputs of this scenario are presented in Table 1. However, the main change between Scenario 2 and 3 is the new architecture presented in Table 3. Figure 9a shows that this new solution improves the performance of the service if compared to Scenario 2. In this case, half of the simulations fix more than 90% of the potential repairs. Additionally, the performance plot in Figure 9b shows that all of the spacecraft perform better under this configuration,this can be seen mainly with the distance to the expected percentage of repairs: in all of the cases, this new scenario achieves closer results to the optimal one. The addition of one servicer to band 8 improved the results in this band, but also increased the ΔV cost of the service, as shown in Figure 10b. Although the drastic increment of ΔV in bands 8 and 12, the global ΔV decreases due to the decrement of all the other ΔVs per transfer, as shown in Figure 10a.

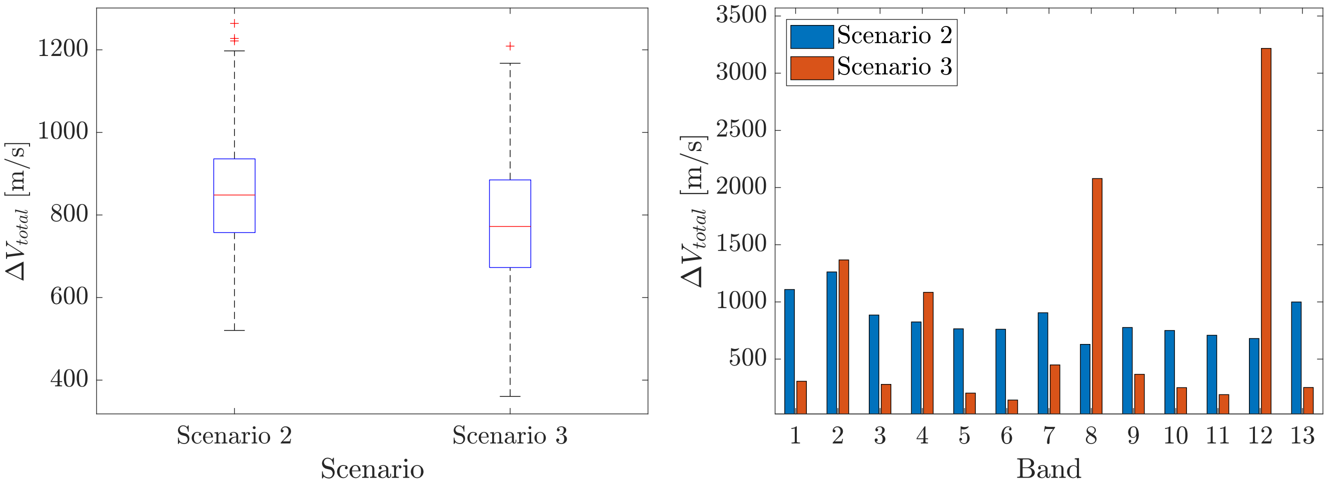
This new architecture can be replicated by setting the $ToF_{max,s}$ to 10 days to lower the occupancy time in case the servicers do not have enough time to perform the repairs in the case of $ToF_{max,s}$ 20 days. However, this solution increases the ΔV cost. It is also worth to note that, in this scenario, the band invasions are lowered due to the decrement in the maximum ΔV per transfer in most of the spacecraft, as shown in Figure 8b.



(a)

(b)

Fig. 9 Repaired satellite of Scenario 2 and Scenario 3 results. In Figure 6a the percentage of failed satellites repaired by the spacecraft and in Figure 6b the band's relative performance compared to the assigned satellite per band.



(a)

(b)

Fig. 10 Total ΔV of Scenario 2 and Scenario 3 results. In Figure 10a the average total ΔV used by the spacecraft while in Figure 10b the average ΔV used by each of the spacecraft.

5 Conclusions

A method to schedule optimal servicing tours for large constellations is presented using tree-search dynamic programming hybridized with beam search. This is then used to analyse the performances of different architectures to conduct the service.

The analysis concludes that as plane changes are expensive in terms of ΔV, each spacecraft should not service a section of the constellation in a Ω-band wider than 4°. Using one servicer per band and a maximum of 20 days per transfer between satellites is enough to keep the entire OneWeb constellation serviced (as it is presented by April 2023) with an average of 90% of potential repairs accomplished. Additionally, lowering the transfer time between satellites to 10 days can be used to lower the servicer occupancy in case the constellation has more satellites per plane or if the service needs to be more recurrent.

Although this methodology and architecture demonstrated to be highly effective with 90% of repairs, it is affected by the different Ω drifts between some planes. This causes changes in bands among the satellites, making them non-serviceable until they enter a new band. It was also noticed that, in some cases, the plane drifts facilitated the movement across bands of the servicers, which could be corrected by tuning the $\Delta V_{max,s}$ individually.

Additionally, the analysis conducted that band jumps are not a viable solution for reducing the number of servicers needed, as the windows generated by the relative movement among the planes are rare, leaving bands unserviced for long periods.

The methodology developed can be applied to other constellations through a procedure analogous to the one delineated in this paper. An examination of the satellite distribution within the constellation is done to define the required number of bands (minimum number of servicers). Subsequently, simulations must be conducted to determine the optimal characteristics for the servicer (E.g. $ToF_{max,s}$ or $\Delta V_{max,s}$), aiming to maximize repairs while minimizing the ΔV used; it is important to ensure that servicers do not reach saturation (occupancy approaching 100%). Fine-tuning at the individual level can increase service performance, particularly in cases where bands exhibit disparate characteristics (E.g. number of satellites or distance between them). Individual tuning is performed by analysing the expected repairs in each band and adjusting the servicer characteristics, to align actual repairs with the expected ones in each band.

Appendix: Summary of the ΔV calculation procedure

In this paper, the ΔV is calculated by considering the effects of the J2 perturbation by using the procedure presented by Shen and Casalino [20]. In this appendix, a brief summary of the procedure is presented for completeness.

Consider a spacecraft that performs a transfer from a satellite k to a satellite k+1, that are lying in different orbital planes of the target constellation. By assuming that the orbital plane drift is provided by the J2 effect only, it is possible to calculate the time needed to perform the transfer between the two orbital planes. This time reads as:

$$t_{opt} = \frac{\Omega_k(t=0) - \Omega_{(k+1)}(t=0) + 2K\pi}{\dot{\Omega}_{(k+1)} - \dot{\Omega}_k} \tag{12}$$

where the integer value of K corresponds to the first opportunity when $t_{opt} > ToF_{max,s}$. Based on this, two scenarios are possible:

- $t_{opt} \leq ToF_{max,s}$: the transfer among the planes with different Ω is possible by exploiting the J2 effect only. Hence, the ΔV is used only to correct semi-major axis, eccentricity and inclination, leading to the following approximated formula:

$$\Delta V/V = \sqrt{(0.5\Delta a/a)^2 + \Delta i^2 + (0.5\Delta e)^2} \tag{13}$$

- $t_{opt} > ToF_{max,s}$: the transfer among the planes with different Ω is not possible. Hence a two-impulse transfer is considered and the resulting ΔV reads as follows:

$$\Delta V = \Delta V_a + \Delta V_b \tag{14}$$

where V_a is the ΔV associated to the first impulse to change the components of $V = [v_x, v_y, v_z]^T$ by fractions s_x, s_y, and s_z, respectively. This reads as:

$$\Delta V_a = \sqrt{(s_x x)^2 + \left(s_y y\right)^2 + (s_z z)^2} \tag{15}$$

and V_b corresponds to the ΔV associated with the second impulse to complete the desired orbital change. This is calculated as:

$$\Delta V_b = \sqrt{(x - s_x x - \Delta x)^2 + \left(y - s_y y\right)^2 + (z - s_z z)^2} \tag{16}$$

where x, y and z read as:

$$\begin{aligned} x &= \left(\Omega_{k+1}\left(t_{k+1}\right) - \Omega_k\left(t_{k+1}\right)\right)\sin i_0 v_0 \\ y &= \frac{a_{k+1} - a_k}{2a_0} v_0 \\ z &= \left(i_{k+1} - i_k\right) v_0 \end{aligned} \tag{17}$$

respectively. The terms a_0, v_0, i_0 and Δx are calculated as follows:

$$\begin{aligned} a_0 &= \frac{a_{k+1} + a_k}{2} \\ i_0 &= \frac{i_{k+1} + i_k}{2} \\ v_0 &= \sqrt{\frac{\mu}{a_0}} \\ \Delta x &= -m s_y y - n s_z z \end{aligned} \tag{18}$$

where $m = \left(7\dot{\Omega}_0\right) \sin i_0 t$, $n = \left(\dot{\Omega}_0 \tan i_0\right) \sin i_0 t$. The optimal Values of s_x, s_y, and s_z that minimize the total ΔV in Equation 14 are computed as follows:

$$s_x x = \frac{2x + my + nz}{\left(4 + m^2 + n^2\right)} \tag{19}$$

$$s_y y = -\frac{2mx - \left(4 + n^2\right) y + mnz}{\left(8 + 2m^2 + 2n^2\right)} \tag{20}$$

$$s_z z = -\frac{2nx + mny - \left(4 + m^2\right) z}{\left(8 + 2m^2 + 2n^2\right)} \tag{21}$$

If small changes in eccentricity are required, the corresponding ΔV is calculated as:

$$\Delta V_e = \frac{1}{2} v_0 \sqrt{\Delta e_y^2 + \Delta e_x^2} \tag{22}$$

Hence, the complete ΔV for this case is:

$$\Delta V' = \sqrt{\Delta V_a^2 + (0.5\Delta V_e)^2} + \sqrt{\Delta V_b^2 + (0.5\Delta V_e)^2} \tag{23}$$

The approximation is well-suited for computing the ΔV of the transfers for the paper under concern, yielding an estimated error lower than 6% [20]. Such an error margin is deemed acceptable for this study. Employing the exact formulation would result in computational times considered to be unacceptably long.

References

[1] C.M. Reynerson. Spacecraft modular architecture design for on-orbit servicing. In *2000 IEEE Aerospace Conference. Proceedings (Cat. No.00TH8484)*, volume 4, pages 227–238, 2000. DOI: 10.1109/AERO.2000.878426.

[2] Joshua P Davis, John P Mayberry, and Jay P Penn. On-orbit servicing: Inspection repair refuel upgrade and assembly of satellites in space. *The Aerospace Corporation, report*, page 25, 2019.

[3] Michael A. Luu and Daniel E. Hastings. Review of on-orbit servicing considerations for low-earth orbit constellations. American Institute of Aeronautics and Astronautics Inc, AIAA, 2021. DOI: 10.2514/6.2021-4207.

[4] Donald J Kessler and Burton G Cour-Palais. Collision frequency of artificial satellites: The creation of a debris belt. *Journal of Geophysical Research: Space Physics*, 83(A6):2637–2646, 1978. DOI: 10.1029/JA083iA06p02637.

[5] Carmen Pardini and Luciano Anselmo. Environmental sustainability of large satellite constellations in low earth orbit. *Acta Astronautica*, 170:27–36, 2020. DOI: 10.1016/j.actaastro.2020.01.016.

[6] Tiago Henrique Matos de Carvalho and Jennifer Kingston. Establishing a framework to explore the servicer-client relationship in on-orbit servicing. *Acta Astronautica*, 153:109–121, 12 2018. DOI: 10.1016/j.actaastro.2018.10.040.

[7] Wen Yao, Xiaoqian Chen, Yiyong Huang, and Michel Van Tooren. On-orbit servicing system assessment and optimization methods based on lifecycle simulation under mixed aleatory and epistemic uncertainties. *Acta Astronautica*, 87:107–126, 2013. DOI: 10.1016/j.actaastro.2013.02.005.

[8] Giacomo Borelli, Gabriella Gaias, and Camilla Colombo. Rendezvous and proximity operations design of an active debris removal service to a large constellation fleet. *Acta Astronautica*, 205:33–46, 4 2023. DOI: 10.1016/j.actaastro.2023.01.021.

[9] Richard Bellman and Robert E Kalaba. *Dynamic programming and modern control theory*, volume 81. Citeseer, 1965.

[10] Andrea Bellome, Joan-Pau Sánchez, Leonard Felicetti, and Stephen Kemble. Multiobjective design of gravity-assist trajectories via graph transcription and dynamic programming. *Journal of Spacecraft and Rockets*, pages 1–19, 2023. DOI: 10.2514/1.A35472.

[11] Dario Izzo, Daniel Hennes, Luís F Simões, and Marcus Märtens. Designing complex interplanetary trajectories for the global trajectory optimization competitions. In *Space Engineering*, pages 151–176. Springer, 2016. DOI: 10.1007/978-3-319-41508-6$_6$.

[12] Dario Izzo. 1st act global trajectory optimisation competition: Problem description and summary of the results. *Acta Astronautica*, 61(9):731–734, 2007. Global Trajectory Optimization. Results of the First Competition Organised by the Advanced Concept Team (ACT) of the European Space Agency (ESA). DOI: 10.1016/j.actaastro.2007.03.003.

[13] Tamas Vinkó and Dario Izzo. Global optimisation heuristics and test problems for preliminary spacecraft trajectory design. *Advanced Concepts Team, ESATR ACT-TNT-MAD-GOHTPPSTD*, 2008.

[14] M. Vasile. A systematic-heuristic approach for space trajectory design. *Annals of the New York Academy of Sciences*, 1017, 2004. DOI: 10.1196/annals.1311.014.

[15] Andrea Bellome. *Trajectory Design of Multi-Target Missions via Graph Transcription and Dynamic Programming*. PhD thesis, Cranfield University, 2023. `https://dspace.lib.cranfield.ac.uk/handle/1826/20830`.

[16] Martin Schlueter, Sven O Erb, Matthias Gerdts, Stephen Kemble, and Jan-J Rückmann. MIDACO on MINLP space applications. *Advances in Space Research*, 51(7):1116–1131, 2013. DOI: 10.1016/j.asr.2012.11.006.

[17] I Michael Ross and Christopher N D'Souza. Hybrid optimal control framework for mission planning. *Journal of Guidance, Control, and Dynamics*, 28(4):686–697, 2005. DOI: 10.2514/1.8285.

[18] Dario Izzo and Marcus Märtens. The Kessler run: on the design of the GTOC9 challenge. *Acta Futura*, 11:11–24, 2018. DOI: 10.5281/zenodo.1139022.

[19] Hong-Xin Shen, Tian-Jiao Zhang, Lorenzo Casalino, and Dario Pastrone. Optimization of active debris removal missions with multiple targets. *Journal of Spacecraft and Rockets*, 55(1):181–189, 2018. DOI: 10.2514/1.A33883.

[20] Hong-Xin Shen and Lorenzo Casalino. Simple ΔV Approximation for Optimization of Debris-to-Debris Transfers. *Journal of Spacecraft and Rockets*, 58(2):575–580, 2021. DOI: 10.2514/1.A34831.

[21] Anastassios E Petropoulos, Daniel Grebow, Drew Jones, Gregory Lantoine, Austin Nicholas, Javier Roa, Juan Senent, Jeffrey Stuart, Nitin Arora, and Thomas Pavlak. Gtoc9: Results from the jet propulsion laboratory team. *Acta Futura*, pages 25–36, 2018. DOI: 10.5281/zenodo.1139152.

[22] William R. Wessels. Use of the weibull versus exponential to model part reliability. In *2007 Annual Reliability and Maintainability Symposium*, pages 131–135, 2007. DOI: 10.1109/RAMS.2007.328115.

[23] John I McCool. *Using the Weibull distribution: reliability, modeling, and inference*, volume 950. John Wiley & Sons, 2012.

[24] Dieter Jungnickel and D Jungnickel. *Graphs, networks and algorithms*, volume 3. Springer, 2005. DOI: 10.1007/978-3-642-32278-5.