Available online at www.sciencedirect.com

ScienceDirect

Advances in Space Research 74 (2024) 3060–3080

ADVANCES IN SPACE RESEARCH (a COSPAR publication)

www.elsevier.com/locate/asr

A constraint programming framework for preliminary mission analysis: Applications for constellation-servicing active debris removal

Adrian Barea [a,*], Juan Luis Gonzalo [b], Camilla Colombo [b], Hodei Urrutxua [a]

[a] Aerospace Systems and Transport Research Group, Universidad Rey Juan Carlos, Fuenlabrada, Spain
[b] Department of Aerospace Science and Technology, Politecnico di Milano, Milan, Italy

Received 20 January 2023; received in revised form 3 July 2024; accepted 5 July 2024
Available online 10 July 2024

Abstract

Constraint Programming is a classical artificial intelligence paradigm characterised by its flexibility for the modelling of complex problems. In the field of space operations, this approach has been successfully used for mission planning and scheduling. This manuscript proposes a framework that leverages the strengths of Constraint Programming for the preliminary analysis of space missions, introducing some modifications to tailor it to the application at hand. Specifically, it uses constraint propagation and search techniques to thoroughly explore the configuration space of a mission in an efficient manner. Consequently, it is able to quantify the performance of precomputed mission choices with respect to the mission requirements, as well as generate new ones that optimise such performance. The proposed methodology has been particularised for two application cases involving active debris removal missions for large constellations in low Earth orbit, namely, a chaser case and a mothership case. The chaser case considers a servicing satellite that rendezvouses with the failed satellites of the constellation and directly transports them to a disposal orbit. The mothership case comprises a servicing satellite that installs deorbiting kits in each of the failed satellites, except for the one removed in the last place. This way, the servicing satellite will only transport this object, while the deorbiting kits will carry out the disposal of the rest of them. This methodology has been successfully used to evaluate a preliminary mission analysis of both application cases developed under ESA's Sunrise project.

Keywords: Active space debris removal; Constellation servicing; Mission analysis; Constraint programming

1. Introduction

The formation of high-density clusters of man-made spaceborne objects poses a significant risk for the sustainability of future space operations. Specifically, it facilitates the occurrence of a collisional cascading effect that would result in an uncontrollable generation of space debris fragments (Kessler and Courpalais, 1978). This is particularly critical for regions of special operational interest, such as Low Earth Orbit (LEO) or Geostationary Orbit, because it could render them unusable for their future utilization. Furthermore, even if the space environment does not reach such critical state, a higher object density entails a potential increase in mission cost and disruptions due to a more frequent necessity of collision avoidance activities (Gonzalo et al., 2021). So as to stabilise the population of spaceborne objects, it is necessary to actively remove several high-impact pieces of debris per year (Liou and Johnson, 2009; Lewis et al., 2012). Hence, active debris removal missions must rigorously select the objects to be removed so that their impact in the space environment is maximised (Barea et al., 2020). In addition to technical and operational considerations, the removal of objects in LEO poses significant legal challenges (Weeden, 2011), including

* Corresponding author.
E-mail addresses: adrian.barea@urjc.es (A. Barea), juanluis.gonzalo@polimi.it (J.L. Gonzalo), camilla.colombo@polimi.it (C. Colombo), hodei.urrutxua@urjc.es (H. Urrutxua).

https://doi.org/10.1016/j.asr.2024.07.015

questions regarding the authority responsible for authorizing and executing the removal of specific objects, which can encompass issues of jurisdiction, ownership, and liability, thus complicating practical implementation of debris removal missions.

Currently, several initiatives to deploy large constellations in the LEO region are being carried out, such as Starlink (Starlink Services LLC, 2022), OneWeb (Network Access Associates Ltd, 2022) and Kuiper (Kuiper Systems LLC, 2022). It is expected that the operation of such constellations will include the end-of-life deorbiting of its defunct satellites. For instance, the Inter-Agency Space Debris Coordination Committee recommends that the objects that terminate their operational phase within the LEO region should be deorbited or transferred to an orbit with an expected residual orbital lifetime of 25 years or shorter (IADC, 2021). More recently, the Federal Communications Commission has officially established a rule stipulating that satellites in the LEO region must deorbit within five years after reaching the end of their mission, as seen in the recent approval of Gen2 Starlink, subject to this regulation (Federal Communications Commission, 2022). Nevertheless, the failure of said disposal processes (either because of a premature failure of a satellite or due to unsuccessful deorbiting manoeuvres) poses a threat, not only for the space environment, but also for the constellation performance. This, along with the presence of the legal issues that the general active debris removal missions face, has motivated the assessment of the feasibility of constellation-servicing debris removal missions (Larbi et al., 2017; Forshaw et al., 2019; Brettle et al., 2021). In particular, the Sunrise project, funded by the European Space Agency (ESA), intends to identify affordable active debris removal strategies for large constellations in LEO. Moreover, this project plans to develop the necessary technologies to perform these missions so as to, eventually, provide a competitive service in the international market. As part of Sunrise, ESA commissioned Phase A studies to different consortia, including one comprising D-Orbit SpA and Politecnico di Milano (Huang et al., 2020; Colombo et al., 2021; Borelli et al., 2021). After the completion of the Phase A studies, the consortium led by Astroscale was chosen to proceed with the next phase of the project (Astroscale, 2022).

As the objects to be removed in constellation-servicing debris removal missions are not known beforehand, the preliminary design of such missions requires an exhaustive analysis of complex mission configurations, especially when dealing with the coordination of several servicing satellites. Constraint programming (Apt, 2003) is a classical artificial intelligence paradigm, characterised by its flexibility for the modelling of complex problems (Pesant et al., 1999). Since its inception, it has proven successful for diverse applications (Wallace, 1996) such as vehicle routing (Shaw, 1998), scheduling (Rodriguez, 2007) and resource allocation (Hladik et al., 2008). In the field of space operations, constraint-based techniques have been extensively used for mission planning and scheduling (Pemberton and Galiber, 2001; Chien et al., 2012), involving applications such as Earth observation (Pemberton, 2000; Frank et al., 2001) and deep space exploration missions (Jiang and Xu, 2017). More specific applications of constraint-based techniques include NASA's EUROPA planning tool (Barreiro et al., 2012), the scientific experiment scheduling of the Rossetta/Philae mission (Simonin et al., 2012) and the mission planning of Orbital Express (Knight et al., 2014).

This work leverages the strengths of Constraint Programming for the preliminary analysis of space missions, introducing some modifications to tailor it to the application at hand. Specifically, the requirements imposed to space missions tend to configure complex search spaces. Consequently, the proposed framework exploits constraint propagation and search techniques to thoroughly explore such spaces in an efficient manner. This way, given a set of predefined mission choices (obtained during a previous mission analysis), the proposed methodology is able to readily quantify their performance with respect to the mission requirements. Then, if a poor performance is shown (or if a previous mission analysis does not exist), the methodology will generate appropriate mission choices so that the desired performance is optimised.

First, Constraint Programming techniques are used to configure a general framework for preliminary mission analysis. One particularity of the proposed framework is that, while Constraint Programming normally focuses on assessing feasibility, constraint bounds including optimality considerations are also introduced. To perform the constraint propagation, best- and worst-case scenarios are defined and solved applying different techniques. Then, that methodology is particularised for the constellation-servicing debris removal mission considered in a Phase A study, developed by the D-Orbit SpA and Politecnico di Milano consortium under ESA's Sunrise project (Huang et al., 2020). In particular, two application cases have been evaluated. Both cases consider a constellation comprising a set of orbital planes with identical inclination, but shifted in Right Ascension of Ascending Node (RAAN). In turn, each of those planes contains a set of satellites that describe an identical circular orbit, but are shifted in angular position within that orbit. Nevertheless, each of the cases uses a different strategy to remove a set of defunct satellites located within the constellation. In the first scenario, the chaser case involves a servicing satellite that rendezvouses with the failed satellites of the constellation and directly transports them to a disposal orbit. In the second scenario, the mothership case comprises a servicing satellite that installs deorbiting kits on each of the failed satellites, except for the one removed in the last place which is deorbited by the servicing satellite itself. This way, the servicing satellite will only transport this object, while the deorbiting kits will carry out the disposal of the rest of them.

The remainder of this manuscript is organized as follows. Section 2 presents the description and mathematical statement of the problem at hand, which is developed in detail in the next sections. Section 3 provides a general description of the proposed Constraint Programming framework. Section 4 particularises the proposed methodology for the chaser application case. Section 5 does the same for the mothership case. Section 6 shows the results that the proposed framework obtains for both application cases. Finally, Section 7 summarises the main conclusions of this work.

2. Problem statement

Let us consider a large satellite constellation in LEO constituted by multiple satellites distributed among several orbital planes. The constellation will be characterized by the inclination and RAAN of each plane, the semimajor axis and eccentricity of the orbits inside each of these planes, and the satellite distribution inside each orbit. For the problem at hand, it is assumed that all the orbital planes have the same inclination and are uniformly distributed in RAAN, and that the orbits are circular with the same semimajor axis and satellites uniformly distributed. The constellation design also includes the implementation of end-of-life disposal processes; however, a certain number of satellites are expected to fail and remain inoperative in orbit. To address this, an active debris removal service for the constellation is introduced.

In the most general case, the active debris removal service is characterized by a servicing spacecraft that rendezvouses with one or more failed constellation satellites, taking actions to ensure their direct re-entry or transporting them to a disposal orbit. Two possible configurations are considered in this work. First, the chaser case, where the servicing spacecraft docks with each failed satellite and transports them to the designated disposal orbit. Second, the mothership case, where the servicing spacecraft attaches deorbiting kits to each failed satellite, except the last one which is deorbited together with the mothership. These two architectures are defined in detail in Sections 4 and 5, respectively.

For a given set of failed satellites to be serviced and a constellation-servicing active debris removal mission architecture, a constrained optimization problem can be formulated to design the mission that removes the largest number of objects. The optimization problem will be characterized by a set of constraints, parameters (or uncontrolled variables), and decision variables (or controlled variables). For both the architectures considered, the mission will have constraints on the maximum mission time Δt^*, and on the maximum fuel mass that can be embarked on the servicer, which will be expressed in terms of maximum ΔV^*:

$$\begin{aligned} \Delta t(params, controlled) \quad &<= \Delta t^* \\ \Delta V(params, controlled) \quad &<= \Delta V^* \end{aligned} \tag{1}$$

Regarding the parameters, in general they will include the constellation design (which is not conditioned by the active debris removal service), the number and distribution of the failed objects, and the chaser design (maximum wet and dry mass). Other variables can be treated as parameters or controlled variables depending on design decisions. In this work, for the chaser case it is assumed that the definition of the injection orbit, the transfer orbit between two constellation planes, and the disposal orbit (expressed in terms on semimajor axis, inclination and eccentricity) are fixed by a previous analysis, and the only controlled variable is the sequence of objects to be removed. For the mothership case, instead, the objects will be removed in monotonic sequence of their position inside the constellation, removing the object sequence as controlled variable, but instead adding as controlled the semimajor axes of the phasing orbits used to transfer between objects within the same constellation plane as well as the inclinations of the initial injection orbit and the drifting orbits. The values for the variables treated as parameter are taken from the mission analysis in (Huang et al., 2020). Note that this choice does not limit the applicability of the method, that can be applied to other values of the fixed parameters. Furthermore, by treating a previously fixed parameter as a controlled variable, it is possible to achieve potential performance improvement with respect to the reference mission analysis. Finally, while for the active debris removal servicing case the number of objects to be visited is typically low, this formulation can be applied to an arbitrary number of objects (only limited by computational capabilities considerations).

The previous problem definition corresponds to maximizing the number of serviced objects for a specific mission. However, during the design of the active debris removal service the goal is not to optimize a specific mission, because the set of failed objects is not known a priori. Instead, the aim in this work is to assess the performance of the mission choices with respect to the mission requirements, quantified through the number of objects N distributed in P planes that can be serviced while complying with the problem constraints. This problem can be formulated as a constraint satisfaction one, obtaining the whole set of feasible controlled and uncontrolled variables for each mission outcome. Then, the problem can be mathematically stated as follows:

$$\begin{aligned} &\text{Given}: \\ &\qquad \mathcal{X} \quad \text{Set of controlled and uncontrolled variables} \\ &\qquad \mathcal{D} \quad \text{Set of domains for each variable in } \mathcal{X} \\ &\qquad \mathcal{C} \quad \text{Set of constraints given by Eq. (1)} \end{aligned}$$

Find the largest $\overline{\mathcal{D}} \subset \mathcal{D}$ such that constraints $\mathcal{C}$ are respected for any $\mathcal{X} \in \overline{\mathcal{D}}$. Note that this differs from a typical Constraint Programming application of finding a feasible solution to a problem (or proving its unfeasibility), in that the complete domain of feasible solutions is sought.

The methodology adopted to tackle this problem is presented in detail in Section 3.

3. Methodology

Given a collection of predefined mission analysis choices, the proposed methodology is able to evaluate the performance of such choices with respect to a series of requirements. That is, the feasibility of achieving different mission outcomes is analysed and the possibility of improving the given mission analysis choices is explored. The aforementioned feasibility depends on controlled as well as uncontrolled variables, with the values of the latter being indeterminate during this mission design phase. Hence, the problem at hand is to obtain the whole set of feasible values of the controlled and uncontrolled variables for each of the mission outcomes. This constitutes a constraint satisfaction problem. A general computational paradigm to deal with this kind of problems is Constraint Programming.

3.1. Constraint Programming resolution process

Constraint satisfaction problems comprise a set of variables, each of them with an associated domain of values, and a set of constraints that relates such values. In turn, a feasible solution of such problems entails a value assignment to every variable, from within their associated domains, such that the whole set of constraints is fulfilled. The main advantage of using Constraint Programming to solve this kind of problems is that it regards constraints as general relations between the domains of the variables, as opposed to other methodologies that consider constraints as analytical mathematical functions. Hence, it provides a great flexibility to develop detailed models of complex problems.

The resolution of a Constraint Programming problem involves the interaction of two different processes, namely constraint propagation and search. The purpose of the constraint propagation process is twofold. First, it checks the feasibility of a given constraint for the considered variable domains (i.e. if there is at least one possible value assignment, from the domains of the considered variables, that fulfills such constraint). Second, it prunes values from the variable domains that cannot appear in a feasible solution.

In general, the use of constraint propagation alone does not guarantee the determination of a feasible solution (or infeasibility) of the problem. However, this can be achieved with the inclusion of an additional search process. This process follows a divide-and-conquer approach to split the variable domains of the original problem, thus partitioning it into several subproblems. The purpose of this technique is to obtain subproblems simple enough so that the constraint propagation process is able to determine their feasibility.

Consequently, the usual workflow of Constraint Programming alternates the constraint propagation and search processes until a feasible solution of one of the subproblems is found or the infeasibility of all the subproblems is demonstrated. However, in the particular case addressed in this work, the whole set of feasible solutions of the problem has to be determined. Therefore, every one of the subproblems has to be demonstrated to be feasible or infeasible.

3.2. Constraint Programming for mission analysis

The general resolution process of Constraint Programming problems has to be tailored to solve the mission analysis problem at hand. As previously stated, this methodology evaluates the feasibility of a set of mission outcomes, henceforth referred as problem instances. Such problem instances can be partitioned into specific intervals of the uncontrolled variables. Thus, the decision variables and their associated domains are characterised by the set of problem instances. In turn, the search process selects the order of evaluation of the problem instances and splits them into simpler ones when required.

The constraints for the constellation-servicing active debris removal problem formulated in Section 2 impose maximum values to specific performance costs. In particular, limits are given to the maximum total time Δt^* and ΔV^* available to carry out the mission, where the ΔV^* limits derive from the dry mass and maximum wet mass of the servicing satellites. The feasibility of a problem instance can then be assessed by finding upper and lower bounds for the costs (i.e., Δt and ΔV) required to implement any mission belonging to said instance, and comparing them to the values of the constraints Δt^* and ΔV^*. If the lower bound of one of the costs is greater than the associated constraint, then no member of that problem instance will be able to satisfy the constraint and the instance is *rejected*. If instead the upper bound of the cost is lower than the constraint, all members of the instance will fulfil the constraint and the instance is *accepted* (at least regarding that cost). Finally, if the constraint value falls within the bounds, some members of the problem instance may be feasible while others may violate the constraint. This instance is then *inconclusive*, and has to be partitioned into smaller ones to assess its feasibility. Sections 4 and 5 detail the models for the computations of the costs Δt and ΔV, and the subproblems defined to compute their upper and lower bounds.

Each of the desired bounds can be obtained solving an optimisation problem, dependent on the controlled and uncontrolled variables. This way, the lower bound can be determined by simply finding the values of the controlled and uncontrolled variables that minimise the performance cost. Likewise, selecting the values of the controlled and uncontrolled variables that maximise the performance cost results in an upper bound. However, albeit simple to obtain, this is not the tightest bound, nor the most logical, because a sound mission analysis is supposed to select the most advantageous values of the controlled variables. Therefore, a tighter upper bound is computed when simultaneously using the uncontrolled variables to maximise the cost and the controlled variables to minimise it. Specifically, the former upper bound represents the worst feasible solution, while the latter stands for the worst optimal solution. Thus they are referred as feasibility and optimality upper bounds, respectively. This way, the constraint propagation problem is reduced to a set of best- and worst-case optimization problems that can be tackled efficiently with well-known optimization techniques. It is important to note that the aim of Constraint Programming is normally to identify feasible problems, and not to optimize performance. In this sense, the inclusion of the tighter optimality bounds represents a particularity of the proposed framework, aimed at providing decision-makers with a holistic view of the trade-offs involved in mission design and execution. From an strict perspective, only the feasibility bounds would be required for a Constraint Programming formulation.

Finally, the domain pruning process exploits the hierarchical relations between the problem instances. That is, if a problem instance is accepted, all the less restrictive instances can be accepted without assessing their feasibility. Similarly, if a problem instance is rejected all the more restrictive instances are rejected.

The resolution process that collects the previous concepts is depicted in Fig. 1. A problem instance is selected

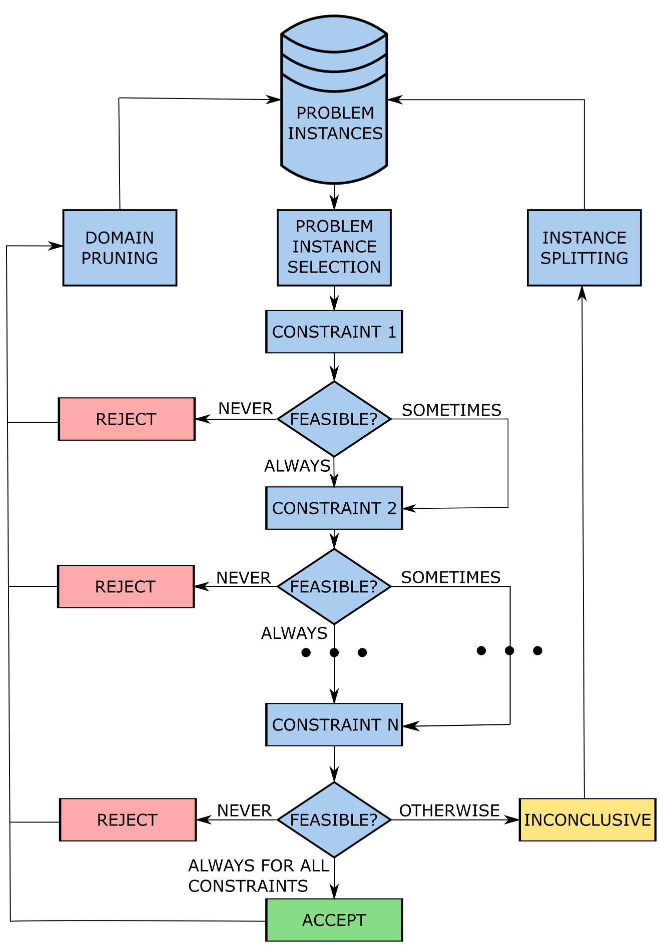

Fig. 1. Problem resolution process.

and its feasibility is sequentially evaluated for each of the constraints. If the instance is rejected by one of the constraints, it is not necessary to evaluate its feasibility for the remaining ones and all the more restrictive problem instances are rejected. In turn, if the instance is deemed inconclusive by one of the constraints, it cannot be accepted by the subsequent constraints (i.e., it will remain inconclusive unless a subsequent constraint rejects it). However, if the instance is still considered inconclusive after the last constraint has been evaluated, it is split into simpler instances. Consequently, those new instances are added to the problem instances set. Finally, if the instance is accepted by every constraint, all the less restrictive problem instances are accepted. This process continues until the whole set of problem instances has been evaluated.

4. Active debris removal mission: Chaser case

The chaser case involves the use of a set of servicing satellites, i.e. chasers, to remove defunct satellites within a constellation. In particular, the defunct satellites are directly transported to a disposal orbit by a chaser. This way, each chaser is assigned a set of orbital planes of the constellation and will perform round trips between those planes and their corresponding disposal orbits until the totality of the defunct objects located within those planes is removed. Specifically, each chaser carries out the following sequence of actions:

1. Remaining in the injection orbit until the RAAN of the first constellation plane is achieved. This RAAN variation is exclusively produced by the nodal drift resulting from the J_2 perturbation.
2. Transferring to the first constellation plane and rendezvousing with the first target.
3. Transferring the first target to its corresponding disposal orbit.
4. Repeating Steps 2 and 3 until the first constellation plane is cleared.
5. Transferring to a drifting orbit and remaining there until the RAAN of the subsequent constellation plane is achieved. Just like in Step 1, this RAAN variation is exclusively produced by the J_2 perturbation.
6. Repeating Steps 2 to 5 until all the constellation planes assigned to the concerning chaser are cleared.
7. The chaser will deorbit together with the last target.

4.1. Predefined mission choices

The proposed methodology will be used to assess the performance of the choices made in previous mission analyses (Huang et al., 2020; Colombo et al., 2021). In particular, such predefined parameters can be classified into the following categories:

Chaser parameters:

- Maximum wet and dry mass of each chaser.
- Semimajor axis, inclination and eccentricity of the initial injection orbit.
- Semimajor axis, inclination and eccentricity of the drifting orbit used to transfer between two different constellation planes.
- Semimajor axis, inclination and eccentricity of the disposal orbit associated to each chaser.

Constellation parameters:

- Defunct satellite mass.
- Semimajor axis, inclination and eccentricity of each constellation plane.
- RAAN difference between two adjacent constellation planes.
- Semimajor axis, inclination and eccentricity of the disposal orbit associated to each defunct satellite.

Thus, regarding the constellation, the uncontrolled variables of the problem at hand are the number of objects to be removed and their position and distribution within the different constellation planes. In turn, the uncontrolled variables related to a chaser are its initial position within the injection orbit as well as its initial relative RAAN with respect to the constellation planes. In addition, the object removal sequence is the only controlled variable. It has to be noted that the predefined mission analysis choices can be readily disregarded and considered as controlled variables so as to achieve potential performance improvements.

Consequently, the initial problem instances can be designated by a tuple (N, P), where N is the number of objects to be removed and P is the number of planes in which these objects are distributed. Those instances can be further partitioned when considering particular object distributions, initial positions of the chaser and RAAN differences between the concerning orbital planes. Moreover, such definition of the initial problem instances shows a clear hierarchical relation between the different instances and, therefore, allows to perform a straightforward domain pruning strategy. Specifically, if an instance (N_0, P_0) is accepted, every instance such that $N \leqslant N_0$ and $P \leqslant P_0$ is instantaneously accepted. In turn, if (N_0, P_0) is rejected, every instance such that $N \geqslant N_0$ and $P \geqslant P_0$ is rejected.

4.2. Feasibility bounds

The requirements imposed by the previous mission analysis involve limitations in the maximum mission time and the available fuel mass. Thus, the values of the aforementioned problem variables have to be optimized so that the most and least advantageous mission time and fuel consumption are obtained. Opportunely, a careful analysis of the problem at hand allows to readily characterise the desired variable values.

The most expensive manoeuvres are the inclination changes performed to drift between the different orbital panes of the constellation. Furthermore, the heavier the chaser, the more fuel-expensive is a manoeuvre. Hence, the sooner such inclination changes are performed, the worse the total fuel consumption will be. Consequently, the object distribution that achieves the feasibility lower bound comprises $N - P + 1$ objects in the first constellation orbital plane and one object in the subsequent planes. Conversely, the feasibility upper bound is achieved when there are $N - P + 1$ objects in the last plane and one object in each of the preceding ones.

The feasibility bounds are directly related with the total RAAN difference traversed by the chaser. Thus, the upper bound is obtained when the last constellation plane to be visited has an initial RAAN very similar to the one of the injection orbit. This way, the chaser has to perform a virtually whole RAAN revolution. In turn, the lower bound is obtained when the RAAN of the injection orbit is identical to the one of the first constellation plane to be visited and the subsequent constellation planes are adjacent to it. Note that the assumption taken for the upper bound, that the final constellation plane visited has a RAAN very similar to the injection plane, is a very conservative one. Indeed, by including in the optimal problem formulation the option to control the RAAN drifting, such a plane could be visited first in the sequence, reducing the feasibility upper bound for time. However, in this study it is decided to work under the simpler, worst-case scenario proposed. Furthermore, because this is the upper limit, this assumption will not lead to the rejection of otherwise feasible problem instances, but increase the number of those classified as inconclusive. An extension of the model to include this additional flexibility can be tackled as a future work.

Finally, the removal sequence and initial situation of the chaser and the objects within their respective orbits are chosen so as to minimise, or maximise, the phasing time necessary to rendezvous with an object when completing a transfer between the injection or disposal orbit and a constellation plane.

4.2.1. Fuel consumption constraint

The fuel consumed during the whole mission can be obtained by iterating in a reverse chronological order (i.e., starting with a chaser with no fuel and adding the fuel consumed during each manoeuvre until the initial mass is retrieved) the following equation:

$$m_i = \left(m_{i+1} + \alpha_i m_{\text{obj}}\right) \exp\left(\frac{\Delta V_i}{g_0 I_{\text{SP}}}\right) - \alpha_i m_{\text{obj}} \tag{2}$$

where i indexes the set of performed manoeuvres (in chronological order), m_i is the mass of the chaser after manoeuvre i, m_{obj} is the mass of a defunct satellite, α_i is 1 if the chaser is transporting a defunct satellite during manoeuvre i (is 0 otherwise), ΔV_i is the ΔV spent during manoeuvre i, g_0 is the gravity acceleration at the Earth's surface and I_{SP} is the specific impulse of the chaser's thruster. This way, given the whole set of ΔV_i values and starting with a m_{i+1} equal to the dry mass of the chaser, Eq. (2) can be iterated to obtain the initial wet mass of the chaser.

Note that Eq. (2) is formulated with the aim of assessing the feasibility of the fuel consumption constraint for a given removal sequence. This is the reason why it is assumed that the final mass is equal to the chaser dry mass. Of course, for a mission with a fixed initial wet mass m_{wet}, a feasible removal sequence would lead to a leftover fuel mass m_{leftover}, which can be computed evaluating the mass equation in chronological order, instead of reverse chronological order, starting from the given m_{wet}. The m_{leftover} would then be the difference between the chaser mass after the final manoeuvre and its dry mass. Furthermore, it is also possible to apply Eq. (2) in cases where a minimum m_{leftover} is imposed (e.g., to account for operational safety margins), by taking as final mass the dry mass plus this m_{leftover}.

Moreover, the manoeuvres are modelled as combined impulses, whose associated ΔV consumption can be computed as:

$$\Delta V = \sqrt{V_1^2 + V_2^2 - 2V_1V_2\cos(\Delta i)} \tag{3}$$

where V_1 and V_2 are the orbital velocities before and after the impulse, respectively, and Δi is the inclination change performed during the manoeuvre.

Finally, it has to be noted that the ΔV consumption does not depend on the initial position or distribution of the objects within the constellation planes. Hence, for a given (N, P) combination, the feasibility bounds for the fuel consumption are obtained when evaluating the aforementioned best and worst object distributions.

4.2.2. Mission time constraint

The mission time can be readily computed by the following expression:

$$\begin{aligned}\Delta t &= \Delta t_{\text{Tdf}} + (N-P+1)\Delta t_{\text{DC}} + (P-1)\Delta t_{\text{DN}} + \\ &+ \Delta t_{\text{RI}} + (N-P)\Delta t_{\text{RC}} + (P-1)\Delta t_{\text{RN}}\end{aligned} \tag{4}$$

where Δt_{Tdf} is the aggregated time spent while coasting in the different drifting orbits, Δt_{RI} is the time elapsed between the departure from the initial injection orbit and the rendezvous with the first object, and Δt_{DC} and Δt_{RC} are, respectively, the time spent to transport an object to its corresponding disposal orbit and the time elapsed between departing such disposal orbit and the rendezvous with the subsequent object; both terms consider that such object is situated in the same constellation plane as the previously removed one. In turn, Δt_{DN} and Δt_{RN} are analogous to the previous terms, but for the case in which the subsequent object is noncoplanar with the formerly removed one.

The time spent while coasting in a drifting orbit is the necessary to nullify the RAAN difference between the target and the drifting orbits, as expressed by the following equation:

$$\Delta t_{\text{df}} = \frac{\Delta\Omega + 2\pi B}{\dot{\Omega}_2 - \dot{\Omega}_{\text{df}}} \tag{5}$$

where Δt_{df} is the drifting time, $\Delta\Omega$ is the initial RAAN difference between the target and the drifting orbits, $B \in \{-1, 0, 1\}$ is a constant chosen so as to obtain the smallest positive Δt_{df}, and $\dot{\Omega}_2$ and $\dot{\Omega}_{\text{df}}$ are the nodal precessions of the target and drifting orbit, respectively. Only the averaged effect of the J_2 perturbation has been considered in the computation of such nodal precessions. Hence, they are represented by the following expression:

$$\dot{\Omega} = -\frac{3nR_{\oplus}^2 J_2}{2p^2}\cos(i) \tag{6}$$

where $R_{\oplus}$ is the equatorial radius of the Earth, J_2 is the coefficient of the spherical harmonic of degree 2 and order 0 of the Earth's gravity field, n is the mean motion of the considered orbit, p is its semilatus rectum and i is its inclination.

Aside from the coasting intervals, the rest of the manoeuvres can be generalised with a two-stage strategy involving a phasing stage and a Hohmann-like transfer with inclination change (although it is not necessary for all of them to comprise a phasing stage or an inclination change). Thus, the remaining terms of Eq. (4) can be modelled as:

$$\Delta\hat{t} = KT_{\text{pha}} + \frac{1}{2}T_{\text{tra}} \tag{7}$$

where $\Delta\hat{t}$ represents a term of Eq. (4) (excluding Δt_{Tdf}), K is the number of revolutions performed by the chaser during the phasing stage and T_{pha} and T_{tra} are the orbital periods of the phasing and transfer orbits, respectively. Moreover, K and T_{pha} have to be chosen so that the mean motion difference between the target and phasing orbits produces a desired difference in true anomaly in an integer number of revolutions:

$$\Delta\theta + 2\pi C + (n_2 - n_{\text{pha}})KT_{\text{pha}} = 0 \tag{8}$$

where $\Delta\theta$ is the difference in true anomaly to be compensated, C is an integer number, and n_2 and n_{pha} are the mean motions of the target and phasing orbits, respectively.

Reformulating Eq. (8) in terms of the orbital periods and isolating K gives:

$$K = \frac{\Delta\theta + 2\pi C}{2\pi\left(1 - \frac{T_{\text{pha}}}{T_2}\right)} \tag{9}$$

Three unknown variables of Eq. (9), namely, K, C and T_{pha}, have to be determined. An initial step to configure an efficient algorithm to solve this equation is to analyse the domain of the concerned variables.

Regarding T_{pha}, in Section 4.2.1, it was stated that the positions of the objects within their constellation plane do not affect the ΔV consumption. It is due to the fact that the phasing stage does not produce a net ΔV consumption, i.e., the ΔV spent to reach the phasing orbit and, subsequently, the transfer orbit is identical to the ΔV that would be spent to directly reach the latter. This entails that the semimajor axis of the phasing orbit must have a value located between the ones corresponding to the semimajor axes of the initial orbit of the current manoeuvre and the transfer orbit. In terms of orbital periods, it means that $T_{\text{pha}} \in [\min(T_1, T_{\text{tra}}), \max(T_1, T_{\text{tra}})]$, where T_1 is the period of the initial orbit of the current manoeuvre.

Regarding K and C, in order to minimise KT_{pha}, K has to be the smallest natural number possible. In addition, for K to be positive, C must have the same sign as $\left(1 - T_{\text{pha}}/T_2\right)$ and its absolute value has to be as small as possible to minimise K. Thus, a lower bound of $|C|$ can be obtained when substituting $K = 1$ as well as the minimum and maximum values of T_{pha} in Eq. (9), and selecting the minimum $|C|$ from the two values obtained. Then, as C must be an integer, if C has to be positive, it is rounded up, otherwise, it is rounded down.

Hence, the aforementioned concepts can be used to configure the following algorithm:

1. The lower bound of C is substituted in Eq. (9).
2. The minimum and maximum values of T_{pha} are introduced in that equation, resulting in the extremes of an interval of possible values of K.
3. If there are natural numbers within such interval, K is the smallest of them.
4. Otherwise, increase $|C|$ in 1 and repeat the previous two steps until K is determined.
5. Once C and K are known, they are substituted in Eq.(9) to obtain T_{pha}.

4.3. Optimality upper bounds

The feasibility upper bounds are obtained when both the controlled and uncontrolled variables are selected to produce the least advantageous value of a particular constraint. However, in practice, the controlled variables will be carefully selected so as to produce the most advantageous results for the mission. The optimality upper bounds take into account this concept by means of the interaction of two antagonistic optimisation processes. Specifically, the uncontrolled variables are chosen to worsen the mission performance, while the controlled ones are selected to improve the constraint value. As a result, the optimality bounds represent the tightest bounds that can be obtained for a particular constraint value, but at a greater computational complexity. As previously indicated, this differs from a classical Constraint Programming application, where the goal is to identify feasible problems without considerations on their optimality.

In this particular case, the only controlled variable to be optimised is the removal sequence. As the fuel consumption constraint does not depend on it, its corresponding feasibility and optimality upper bounds are identical. In turn, the mission time constraint depends on the removal sequence so the optimality bounds have to be computed.

In addition, the uncontrolled variables to be considered during this computation are the initial positions of the objects and the chaser as well as the aggregated RAAN traversed by the chaser while drifting. It has to be noted that the time elapsed between the removal of two consecutive objects does not depend on their absolute initial positions, but on their relative geometry (i.e., the elapsed time would be the same if a constant is added to the initial position of the two objects). It entails that this problem can be decomposed into smaller ones that account for the initial position of the chaser and the objects in a single constellation plane. Then, these plane-wise individual solutions can be connected by adding a constant quantity to the whole set of positions corresponding to each solution, so that the final position of one solution corresponds to the initial position of the subsequent one.

When matching two consecutive solutions, it is important to note that the situation for the chaser is slightly more complex than for the objects to be removed. Indeed, the constant quantity that is added to the initial position of the objects in the plane should be added to the initial position of the chaser inside that particular plane (i.e., before any phasing manoeuvre inside the plane), not to its overall initial position. The difference between both will be due to the drifting and transfer manoeuvres performed before arriving to that plane. Therefore, to strictly compute the quantity that needs to be added to the initial position of the chaser, the transfers and drift would need to be propagated backwards starting from the position in the plane. However, this process can be circumvented by directly considering such chaser position within the constellation plane during the problem resolution. As a consequence, the aggregated RAAN traversed by the chaser while drifting can be uncloupled from this problem. Thus obtaining the optimality upper bound by simply considering the largest possible traversed RAAN.

All in all, each of those subproblems involve an unconstrained Bilevel Mixed Integer Nonlinear Programming problem, represented by the following objective function:

$$\max_{\mathbf{u}} \left\{ \min_{\mathbf{X}} \{ \Delta t(\mathbf{u}, \mathbf{X}) \} \right\} \tag{10}$$

where $\mathbf{X}$ is a matrix of binary variables that represents the removal sequence and $\mathbf{u}$ is the vector of initial arguments of latitude of the chaser and the objects, which represents their respective initial positions within their orbits.

The resolution process of this problem follows the structure shown in Fig. 2. The upper lever, hereinafter referred to as parameter search, explores $\mathbf{u}$ and sends promising values to the lower lever. In turn, the lower level obtains the optimal removal sequence for the received $\mathbf{u}$ and provides the resulting Δt to the upper level. This way, the upper level can use the Δt information to select a subsequent $\mathbf{u}$ that can potentially produce a worse Δt.

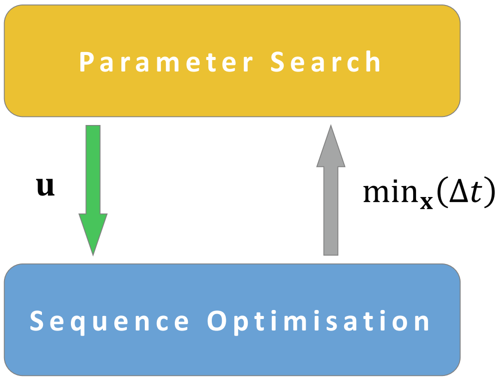

Fig. 2. Optimality upper bound process.

4.3.1. Parameter search

Assuming that u_1 represents the position of the chaser, it can be seen that every permutation of the elements of $\mathbf{u}$ (aside from u_1) represents the same problem. Hence, eliminating that permutation symmetry results in a reduction of the search space size by a factor of $N!$. It can be readily achieved by ordering the arguments of latitude in a monotonic way, as imposed by the following constraint:

$$u_i \leqslant u_{i+1} \quad \forall i : 2 \leqslant i \leqslant N \tag{11}$$

where i indexes the components of $\mathbf{u}$.

Therefore, considering the sequence optimisation problem as a black-box function, the parameter search problem involves the maximisation of the objective function defined by Eq. (10), subject to Eq. (11). As Eq. (10) is non-smooth and discontinuous, derivative-free methodologies are proposed to solve this problem. In addition, said function is computationally expensive to evaluate because it requires the resolution of an Integer Programming problem. This arises a dilemma about the trade-off between objective function evaluations and solution quality. As a result, two different methodologies are proposed to solve the parameter search problem.

On the one hand, Generalised Pattern Search (Lewis and Torczon, 2000) constitutes a derivative-free direct search methodology that follows a similar strategy to steepest descent approaches. This way, a single solution is iteratively improved by means of sampling its neighbouring points. Thus being prone to obtaining disadvantageous local optimal solutions and, consequently, being sensitive to the initial guess used to initialise the search. On the other hand, Evolutionary Algorithms provide a better search space exploration by means of maintaining a population of possible solutions, but at the cost of a considerably larger number of objective function evaluations.

The selection of a good-quality initial guess not only is of capital importance for the performance of the Generalised Pattern Search method, but also can have a favourable effect when including it within the initial population of an Evolutionary Algorithm. This phenomenon is going to be quantified by means of the use of two possible initial guesses.

The most straightforward of them is the one that results in the feasibility upper bound. This solution provides an artificial selection of the removal sequence to maximise the mission time. This implies that a simple modification of the sequence can result in notable improvements of such time. Hence, the $\mathbf{u}$ provided by this solution is very unlikely to be the global optimum.

The other considered initial guess involves all the objects with an identical orbital position, such that, the combination of this position and the one selected for the chaser maximise Δt_{RI}. This kind of solution makes it impossible to improve the mission time by modifying the removal sequence and, despite being an unrealistic and degenerate case, is very likely to be near-optimal, or even the global optimum, if the ratio $\Delta t_{\mathrm{RC}}(\Delta u = 0)/\max(\Delta t_{\mathrm{RC}})$ is close to 1, where $\max(\Delta t_{\mathrm{RC}})$ is the maximum Δt_{RC} that can be achieved across the domain of possible $\mathbf{u}$ vectors.

4.3.2. Sequence optimisation

The sequence optimisation problem involves the determination of the removal sequence, as well as the selection of the objects to be removed in case it is not required to remove all of them. As the initial arguments of latitude of the different objects are obtained by the parameter search problem, the Δt spent during each of the possible transfers can be unambiguously computed and the sequence optimisation problem can be formulated as an Integer Linear Programming problem. In particular, this problem intends to minimise the total Δt spent during the mission:

$$\min\left\{\sum_{i\in D}\sum_{\substack{j\in D\\ i\neq j}}\Delta t_{ij}X_{ij}\right\} \tag{12}$$

where D is the set of possible objects to be removed (indexed by i and j), X_{ij} is a binary variable that is 1 if and only if a transfer from object i to j is performed and Δt_{ij} is the time elapsed during each of those transfers.

Considering a directed graph formed by D and the different transfers between those objects, the removal sequence can be modelled as a directed cycle comprising the selected objects and an additional dummy object (labeled as object 1). That object represents both the chaser initial and final state, i.e., its outgoing edge represents the transfer between the injection orbit and the first object, while its incoming edge represents the final transfer of the chaser to its corresponding disposal orbit. Hence, a set of linear constraints that guarantees the formation of such cycle is subsequently defined.

First, a transfer between an object and itself cannot be performed:

$$X_{ii} = 0 \quad \forall i \in D \tag{13}$$

Moreover, if an object is removed, it has exactly one outgoing transfer. Otherwise, it cannot have such transfers:

$$\sum_{\substack{j\in D\\ i\neq j}} X_{ij} = Y_i \quad \forall i \in D \tag{14}$$

where Y_i is a binary variable that is 1 if and only if the object i is removed.

Likewise, each removed object has exactly one incoming transfer:

$$\sum_{\substack{j\in D\\ i\neq j}} X_{ji} = Y_i \quad \forall i \in D \tag{15}$$

Furthermore, the number of selected objects is equal to the number of objects required to be removed (N) plus the dummy object:

$$\sum_{i\in D} Y_i = N + 1 \quad \forall i \in D \tag{16}$$

Note that if the whole set of objects has to be removed, case known as Travelling Salesman Problem (TSP) (Dantzig et al., 1954), this latter constraint is equivalent to imposing that $Y_i = 1\ \forall i \in D$.

The aforementioned set of constraints does not prevent the appearance of disjoint cycles, also known as subtours. The reason for their appearance is that they allow to substitute expensive transfers of the main cycle for cheaper ones that form subtours. This phenomenon is specially impactful for problems with lots of potential advantageous subtours, such as problems that involve a large candidate object pool or objects distributed in several orbital planes. Thus, it is important to select the most adequate methodology to deal with this problem.

A common strategy to eliminate potential subtours entails the consideration of additional ad hoc constraints of the following form (Dantzig et al., 1954):

$$\sum_{i\in R}\sum_{\substack{j\in R\\ i\neq j}} X_{ij} \leqslant |R| - 1 \tag{17}$$

where R is the set of objects that form a subtour. The number of possible subtours of length $|R|$ corresponds to the number of possible $|R|$-combinations of objects divided by $|R|$, i.e., $|D|!/(|R|(|D| - |R|)!)$. Hence, the straightforward use of this strategy would result in an impractical number of constraints, even for problems of moderate size. Consequently, two different approaches can be considered as a function of the problem size.

On the one hand, for problems involving five or less objects, the addition of Constraints (17) particularised for two-object subtours (i.e. round trips between two objects) is enough to prevent the appearance of subtours. Such constraints can be defined as follows:

$$X_{ij} + X_{ji} \leqslant 1 \quad \forall i \in D \quad \forall j \in D : (i < j) \tag{18}$$

On the other hand, for larger problems, Constraints (17) can be dynamically generated during the resolution of the problem. This way, if a solution with subtours is found, the solution will be rejected and the corresponding subtour elimination constraints will be added to the model. This strategy is usually very efficient, but its performance significantly degrades for problems with a large number of relevant subtours, specially for instances involving objects distributed within different orbital planes.

It has to be noted that the considered variables do not store information about the removal sequence (i.e., the order in which each of the transfers is performed). However, the appearance of subtours can be prevented by means of unambiguously defining such removal sequence. Hence, no-subtour formulations can be configured by means of the inclusion of additional variables.

Such formulations have been thoroughly explored for the Travelling Salesman Problem (Langevin et al., 1990). The most notable of them (Miller et al., 1960) involves the introduction of additional variables z_i that directly determine the order in which each node is visited. Assuming that $z_1 = 0$, i.e., the chaser starts at the node 1, the domain of the rest of those variables is $z_i \in [1, |D| - 1]$ and their corresponding order can be obtained with the following set of constraints:

$$z_i - z_j + |D|X_{ij} \leqslant |D| - 1 \quad \forall i \in D : (i > 1), \quad \forall j \in D : ((j > 1) \wedge (i \neq j)) \tag{19}$$

The TSP no-subtour formulations achieve a weaker linear relaxation than the one that would be obtained with the use of Constraints (17) or require a large number of constraints (Langevin et al., 1990). Thus, the dynamic elimination is often preferred. However, they can be useful when there is a large number of relevant subtours.

A more general no-subtour formulation, suitable for problems in which it is not necessary to remove all the candidate objects, was proposed in Barea et al. (2020). It includes the sequence information within the X_{ij} variables, transforming them into the new X_{ijk} variables and substituting each X_{ij} instance for the expression $\sum_{k\in K} X_{ijk}$. This way, X_{ijk} is 1 if and only if the k-th transfer is performed between objects i and j. Moreover, one transfer corresponds to each position k:

$$\sum_{i\in D}\sum_{\substack{j\in D\\ i\neq j}} X_{ijk} = 1 \qquad \forall k \in [1, N + 1] \tag{20}$$

Furthermore, the first and last transfers correspond to the departure from the injection orbit and the disposal of the chaser, respectively:

$$\sum_{j\in D} X_{1j1} = 1; \qquad \sum_{j\in D} X_{j1(N+1)} = 1 \tag{21}$$

Finally, the subtour appearance can be readily prevented by imposing that the final object of a transfer is the first of the subsequent one:

$$\sum_{\substack{i \in D \\ i \neq j}} X_{ijk} = \sum_{\substack{i \in D \\ i \neq j}} X_{ji(k+1)} \quad \forall j \in D, \quad \forall k \in [1, N] \tag{22}$$

This formulation is especially tailored for instances in which the candidate object pool is large, but only a small subset of those objects must be removed. Hence outperforming the dynamic elimination strategy in those instances. However, its performance deteriorates when increasing the number of objects to be removed.

Table 1 summarises the recommended formulation selection for different kinds of problem instances. Note that it is not clear which formulation to use for cases that involve a large set of candidate objects, distributed among different orbital planes, such that a large subset of those objects is required to be removed. The reason for that is the degradation of the performance of both the dynamic elimination strategy and the general no-subtour formulation for such kind of instances.

5. Active debris removal mission: Mothership case

The mothership case, just like the chaser case, involves the use of a set of servicing satellites, i.e., motherships, to remove defunct satellites within a constellation. However, in this case, the defunct satellites are transported to their corresponding disposal orbits by deorbiting kits, which have been previously attached to them by a mothership. This way, each mothership rendezvous with each of its assigned objects to deploy the deorbiting kits and will only transfer to a disposal orbit when performing the removal of its last associated object. During that manoeuvre, the mothership will transport such object to the disposal orbit so that both can simultaneously reenter the atmosphere. Hence, the sequence of actions carried out by each mothership is analogous to the one explained in Section 4, save for the intermediate transfers to the disposal orbit.

5.1. Predefined mission choices

Analogously to the chaser case, the following parameters are extracted from a previous mission analysis.

Mothership parameters:

- Maximum wet and dry mass of each mothership.
- Mass of the deorbiting kits.
- Semimajor axis, inclination and eccentricity of the initial injection orbit.
- RAAN difference between the initial injection orbit and the target constellation plane.
- Semimajor axis, inclination and eccentricity of the drifting orbit used to transfer between two different constellation planes.
- Semimajor axis, inclination and eccentricity of the phasing orbit used to transfer between two objects within the same constellation plane. Note that such phasing orbit should keep a minimum safety distance from the concerned constellation orbit, so as not to generate a risk of potential collisions of the mothership with active satellites located in that constellation plane.
- Semimajor axis, inclination and eccentricity of the disposal orbit associated to each mothership.

Constellation parameters:

- Defunct satellite mass.
- Semimajor axis, inclination and eccentricity of each constellation plane.
- RAAN difference between two adjacent constellation planes.

The uncontrolled variables, save for the RAAN difference between the different orbital planes, are analogous to the ones of the chaser case. The same is true for the problem instances and domain pruning strategy. However, in this case, it is logical to remove the defunct objects in order of monotonically increasing (or decreasing) arguments of latitude. Therefore, there would be no controlled variables and the feasibility and optimality bounds would be identical. This provides a good opportunity to disregard some of the predefined choices, thus potentially improving the performance of the mission. Specifically, the semimajor axes of the phasing orbits used to transfer between two objects within the same constellation plane as well as the inclinations of the initial injection orbit and the drifting orbits are considered as the controlled variables of the problem at hand.

5.2. Feasibility bounds

The requirements imposed by the previous mission analysis involve limitations in the maximum mission time and the ΔV consumed during the mission. Therefore, the most

Table 1
MIP formulation selection criteria.

$\|D\|$	Object distribution	N	Formulation
$\leqslant 5$	Any	$[1, \|D\|]$	Two-object constraints (Dantzig et al., 1954)
Small	Any	$[1, \|D\|]$	Dynamic elimination (Dantzig et al., 1954)
Large	Coplanar	$\|D\|$	Dynamic elimination (Dantzig et al., 1954)
Large	Noncoplanar	$\|D\|$	TSP No-subtour (Langevin et al., 1990; Miller et al., 1960)
Large	Any	$\ll \|D\|$	General No-subtour (Barea et al., 2020)
Large	Noncoplanar	$\approx \|D\|$	Undetermined

and least advantageous values of the object distribution within the different constellation planes and the initial positions of the different objects have to be determined.

The feasibility lower bound is obtained when the initial positions of every object are such that the phasing is achieved when transferring to the predefined phasing orbit (recall that the phasing orbit does not intersect the orbit of the constellation plane) and, upon arrival, instantaneously transferring back to the constellation orbit, i.e., the phasing is directly achieved by the transfer orbit.

In turn, the upper bound also depends on the object distribution. Specifically, the objects have to be allocated within the constellation planes such that the aggregated argument of latitude compensated by the whole set of phasing manoeuvres is maximised. A naive approach would be to consider that a complete revolution in argument of latitude is compensated for each orbital plane. However, a tighter feasibility bound can be achieved without a meaningful computational effort.

First, it is assumed that the objects are equally spaced within its plane. Otherwise, the first object to be removed could be regarded as a controlled variable and one of the objects adjacent to the largest gap in argument of latitude would be assigned to it. This would result in a better solution, but not necessarily an upper bound of the feasible solution set.

Second, it is postulated that, due to the accumulation of uncertainties during the clearance of former planes, the mothership arrives to the second and subsequent constellation planes with a phasing error. This way, the arrival is produced at an argument of latitude equidistant from two objects and an additional phasing manoeuvre has to be performed to correct it.

Taking into account both assumptions, the object distribution can be obtained by minimising the argument of latitude not compensated by the phasing for each constellation plane:

$$\min \left\{ \frac{2\pi}{N\prime_1} + \sum_{\substack{p\in P\\ p>1}} \frac{\pi}{N\prime_p} \right\} \tag{23}$$

where $N\prime_p$ is the number of objects allocated to the plane $p \in P$. Evidently, the total number of allocated objects has to be the number of objects to be removed:

$$\sum_{p\in P} N\prime_p = N \tag{24}$$

Moreover, $N\prime_p$ must be natural numbers. However, when relaxing such integrality condition, the problem defined by Eqs. (23, 24) has the following analytical solution:

$$N\prime_1 = \frac{N\sqrt{2}}{P-1+\sqrt{2}}; \qquad N\prime_p = \frac{N}{P-1+\sqrt{2}} \quad \forall p \in P : p > 1 \tag{25}$$

Then, two possible integer solutions can be obtained, resulting from rounding up or down the value of $N\prime_1$. The rest of variables can be rounded so that Eq. (24) is fulfilled and both solutions are computed to select the one that minimises Eq. (23).

5.2.1. ΔV constraint

The ΔV spent during the mission can be readily obtained from the following expression:

$$\Delta V = \Delta V_{\mathrm{RI}} + (N-P)\Delta V_{\mathrm{RC}} + (P-1)\Delta V_{\mathrm{RN}} + \Delta V_{\mathrm{D}} \tag{26}$$

where the subindices RI, RC and RN represent the ΔV spent to rendezvous with the next object to be removed if it is the first object in the sequence, coplanar with the previously removed object or noncoplanar with it, respectively. Moreover, ΔV_D represents the ΔV used to transfer the mothership, along with the last object of the sequence, to its corresponding disposal orbit.

Each of the terms of the right hand side of Eq. (26) comprises several impulses (modelled using Eq. (3)) arranged to configure Hohmann-like transfers with inclination changes as well as intermediate phasing and drifting orbits. The semimajor axis and inclination changes performed during each of those impulses are optimised so that their corresponding term is minimised, while complying with the predefined mission choices.

It has to be noted that the predefined mission choices, along with the optimised terms of the right hand side of Eq. (26), unambiguously define the spent ΔV for each (N, P) tuple. Hence, both feasibility bounds associated to the ΔV constraint collapse into a single quantity regardless of the values of the uncontrolled variables.

5.2.2. Mission time constraint

The total mission time can be computed with the following equation:

$$\Delta t = \Delta t_{\mathrm{Tdf}} + \Delta t_{\mathrm{RI}} + (N-P)\Delta t_{\mathrm{RC}} + (P-1)\Delta t_{\mathrm{RN}} \tag{27}$$

where Δt_{Tdf} is the aggregated time spent while coasting in the different drifting orbits and the subindices RI, RC and RN are analogous to the ones found in Eq. (26), but applied to the transfer time. Each of the individual drifting times included in Δt_{Tdf} can be obtained using Eq. (5). In turn, the rest of the Δt components of Eq. (27) can be obtained by means of the resolution of Eq. (7).

Unlike the ΔV constraint, the mission time depends on the object distribution as well as the initial positions of the mothership and the objects to be removed. Hence, the feasibility bounds are achieved when considering the most and least advantageous values of those uncontrolled variables.

5.3. Optimality bounds

As the removal sequence is predefined, this problem does not have controlled variables and the optimality bounds correspond to the feasibility bounds. However, so as to potentially improve the performance of the mission, the semimajor axes of the phasing orbits as well as the inclinations of the initial injection orbit and the drifting orbits are considered as controlled variables of the problem. Moreover, the considered uncontrolled variables are the object distribution within the different constellation planes and the initial positions of the different objects.

Despite the similarities of the present problem with the chaser case, its structure and the resolution methods applied to it are radically different. Specifically, this problem is also a Mixed Integer Nonlinear Programming problem because the semimajor axes have to be chosen such that the phasing orbits perform an integer number of revolutions. Nevertheless, its resolution can be decomposed into two sequential phases. The first phase allows the selection of phasing orbits with fractional numbers of revolutions, thus constituting a Nonlinear Programming problem. Then, the second phase corrects the solution to achieve the revolution integrality, resulting in an Integer Linear Programming problem.

5.3.1. Phasing and drifting orbit optimisation

The first phase to obtain the optimality bounds involves the selection of the semimajor axes of the phasing orbits as well as the inclinations of the initial injection orbit and the drifting orbits so that the mission time is minimised, as shown in the following objective function:

$$\min \left\{ \sum_{k \in K} \Delta t_{\text{pha}}(a_k) + \sum_{\ell \in L} \Delta t_{\text{df}}(i_\ell) \right\} \tag{28}$$

where K represents the set of phasing manoeuvres (indexed by k), L is the set of drifting orbits (indexed by ℓ), Δt_{pha} is the time spent during the phasing manoeuvres, Δt_{df} is the time elapsed while drifting, a_k stands for the semimajor axis of the phasing orbit k and i_ℓ is the inclination of the drifting orbit ℓ.

Specifically, this problem entails the redistribution of the ΔV available for phasing and drifting manoeuvres to achieve the desired solution. This is modelled by the following constraint:

$$\sum_{k \in K} \Delta V_{\text{pha}}(a_k) + \sum_{\ell \in L} \Delta V_{\text{df}}(i_\ell) = \Delta V^* \tag{29}$$

where ΔV_{pha} and ΔV_{df} represent the ΔV spent to achieve the phasing and drifting orbits, respectively, and ΔV^* is the ΔV available for the considered manoeuvres.

This problem can be readily solved with conventional Nonlinear Programming techniques. However, the dual-based methodology proposed in Barea et al. (2022) provides an efficient way to obtain the global optimum of the problem. For the sake of completeness, the application of such methodology to the problem at hand is subsequently explained in a concise manner.

The Lagrangian function of the problem at hand can be defined as:

$$\mathcal{L} = \sum_{k \in K} \Delta t_{\text{pha}}(a_k) + \sum_{\ell \in L} \Delta t_{\text{df}}(i_\ell) - \lambda \left(\sum_{k \in K} \Delta V_{\text{pha}}(a_k) + \sum_{\ell \in L} \Delta V_{\text{df}}(i_\ell) - \Delta V^* \right) \tag{30}$$

where λ is the dual variable associated to Eq. (29).

The optimality conditions of this problem can be obtained by means of nullifying the gradient of the Lagrangian function, yielding:

$$\frac{\mathrm{d}\Delta t_{\text{pha}}}{\mathrm{d}a_k}(a_k) - \lambda \frac{\mathrm{d}\Delta V_{\text{pha}}}{\mathrm{d}a_k}(a_k) = 0 \quad \forall k \in K \tag{31a}$$

$$\frac{\mathrm{d}\Delta t_{\text{df}}}{\mathrm{d}i_\ell}(i_\ell) - \lambda \frac{\mathrm{d}\Delta V_{\text{df}}}{\mathrm{d}i_\ell}(i_\ell) = 0 \quad \forall \ell \in L \tag{31b}$$

The derivative chain rule can be used to isolate λ in Eqs. (31), resulting in the following conservation law:

$$\lambda = \frac{\mathrm{d}\Delta t_{\text{pha}}}{\mathrm{d}\Delta V_{\text{pha}}}(a_k) = \frac{\mathrm{d}\Delta t_{\text{df}}}{\mathrm{d}\Delta V_{\text{df}}}(i_\ell) \qquad \forall k \in K, \quad \forall \ell \in L \tag{32}$$

Eqs. (32) can be inverted and substituted into Eq. (29) to configure the following univariate function:

$$\phi(\lambda) = \sum_{k \in K} \Delta V_{\text{pha}}(a_k(\lambda)) + \sum_{\ell \in L} \Delta V_{\text{df}}(i_\ell(\lambda)) - \Delta V^* \tag{33}$$

A root of Eq. (33) automatically fulfills the optimality conditions, i.e., Eqs. (29, 32). Hence, the solution of the problem simply involves the determination of λ, regardless of the number of phasing and drifting orbits. Then, the values of a_k and i_ℓ can be retrieved with the inverse of Eqs. (32).

5.3.2. Phasing orbit correction

The previous problem regards a_k as a continuous variable. However, only the values that result in an integer number of revolutions of the phasing orbits are feasible. A feasible solution can be obtained by computing the number of revolutions associated to each a_k, rounding up such number and computing the corrected values of each a_k. This gives a solution with a greater mission time, but a lower ΔV consumption. Therefore, considering the solution with rounded up revolutions as the reference, the problem at hand involves deciding which revolutions to round up or down such that the ΔV surplus can be optimally redistributed, hence improving the mission time. This can be modelled with the following objective function:

$$\min \sum_{k \in K} \Delta(\Delta t_k) \Psi_k \tag{34}$$

where $\Delta(\Delta t_k)$ is the Δt difference between the rounded down and the rounded up revolutions and Ψ_k is a binary variable that is 1 if the revolutions associated to the k-th phasing orbit are rounded down. The following constraint models the ΔV redistribution:

$$\sum_{k \in K} \Delta(\Delta V_k) \Psi_k \leqslant \Delta V\prime \tag{35}$$

where $\Delta(\Delta V_k)$ is the ΔV difference between the rounded down and the rounded up revolutions and $\Delta V\prime$ is the ΔV

surplus to redistribute. Eqs. (34, 35) define an Integer Linear Programming problem that can be readily solved with branch-and-bound methods.

6. Results

The mission analysis from Huang et al. (2020) is subsequently compared with the results obtained with the proposed methodology. Specifically, predefined performance baselines are defined for both application cases and the achieved level of fulfilment of such baselines is discussed. Table 2 shows the predefined mission parameters shared by both application cases.

6.1. Chaser case

Table 3 shows the requirements imposed for the chaser case. In addition, its associated performance baseline involves removing three objects located in different orbital planes. Moreover, it is assumed that the chaser will be deployed with a rideshare launch. Thus, the initial RAAN associated to the injection orbit is considered as an uncontrolled variable. This implies the analysis of two possible mission geometries: one case with *Type 1 Positioning*, which considers that such initial RAAN is not included within the interval defined by the constellation planes to be cleared; and another case with the *Type 2 Positioning*, which involves an initial RAAN inside that interval, resulting in a more disadvantageous case.

Tables 4 and 5 summarise the mission analysis carried out in Huang et al. (2020). In particular, the inclination of the drifting orbits depends on the initial positioning of the chaser. In turn, the disposal orbit associated to an object depends on the constellation plane in which the next object to be removed is located, as well as the initial positioning of the chaser.

Consequently, according to Huang et al. (2020), the aforementioned parameter selection makes it possible for the chaser to remove three coplanar objects within the required mission time, regardless of the initial positioning of the chaser. However, in order to remove three noncoplanar objects, it would be necessary to increase the wet mass of the chaser by 5 and 35 kg for the initial positioning of Type 1 and 2, respectively.

Table 2
Predefined parameters for both application cases.

Parameter	Value
Object mass (kg)	150
Constellation altitude (km)	1200
Constellation inclination (deg)	87.9
Constellation eccentricity	0
$\Delta\Omega$ between adjacent planes (deg)	15.2
Number of constellation planes	12
Injection orbit altitude (km)	500
Injection orbit inclination (deg)	86
Injection orbit eccentricity	0

Table 3
Mission requirements for the chaser case.

Parameter	Value
Maximum mission time (years)	5
Dry mass of the chaser (kg)	245
Maximum wet mass of the chaser (kg)	520
Specific impulse of the chaser (s)	285

Table 4
Drifting orbit parameters for the chaser case.

Parameter	Positioning 1	Positioning 2
Perigee altitude (km)	500	500
Apogee altitude (km)	1100	1100
Inclination (deg)	87.1082	86.5896

Hence, the results of the proposed methodology are compared with such conclusions. The first step in its application is the configuration of the problem instances. Specifically, as the (N, P) combinations are going to be evaluated, an upper bound of the maximum N to be evaluated has to be obtained. Such maximum number of objects can be readily computed by iterating backwards Eq. (2), starting from the dry mass of the chaser, until a mass of the chaser greater than the maximum wet mass is achieved. This yields that, under the imposed requirements, it is impossible to remove more than three objects with a single mission.

Consequently, Fig. 3 shows the results of computing the feasibility bounds for the (N, P) problem instances of up to three objects, where the green instances are feasible in any case, the red ones are always infeasible and the yellow ones are inconclusive. The evaluated instances are highlighted with black squares, while the feasibility of the rest was concluded with constraint propagation. So far, the feasibility showed in Fig. 3 is in line with the conclusions of Huang et al. (2020). That is, for Type 2 positioning, it is only possible to remove three objects if they are coplanar. In turn, for Type 1 positioning, there are cases in which it is not feasible to remove three noncoplanar objects. However, the mission time upper feasibility bound of the $(3, 3)$ instance, for Type 1 positioning, violates the maximum mission time by just 2.8 h. Moreover, further splitting such problem instance, it is determined that it is feasible for any aggregated $\Delta\Omega$ traversed by the chaser with a value lower than 359.961 degrees. Thus, unlike Huang et al. (2020) proposes, it is not worth to modify the design of the chaser for such a small and improbable violation of the maximum mission time.

The optimality bounds do not provide additional information about the problem feasibility, as the $(3, 3)$ instance has a predefined removal sequence, i.e., clearing the planes in a monotonic RAAN order. However, such analysis can provide insightful information about the influence of sequence optimisation for problems with a similar structure, as well as about the performance of the proposed

Table 5
Disposal orbit parameters for the chaser case.

Parameter	Coplanar	Noncoplanar Pos. 1	Noncoplanar Pos. 2
Perigee altitude (km)	351.422	351.4722	351.4959
Apogee altitude (km)	1100	1100	1100
Inclination (deg)	87.9	87.1082	86.5896

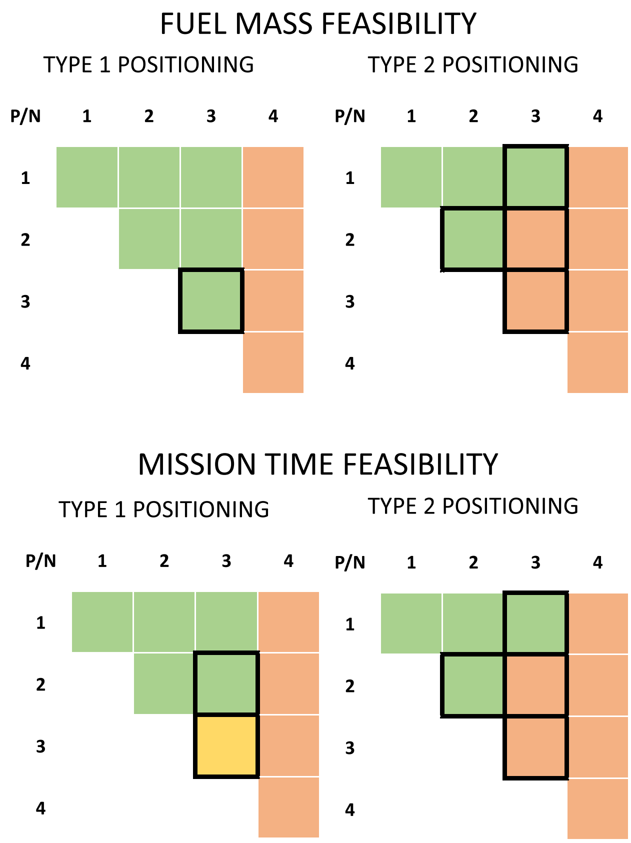

Fig. 3. (N, P) instances diagram for the chaser case.

techniques to solve this kind of Bilevel Mixed Integer Nonlinear Programming problems.

As the considered sequence involves at most four objects (three objects to be removed plus the dummy object), the formulation with two-object constraints is used for the sequence optimisation level. Regarding the parameter search level, it is solved using the two methodologies mentioned in Section 4.3.1 (Generalised Pattern Search and an Evolutionary Algorithm), as well as the two initial guesses discussed there (i.e., the object distribution that gives worst feasible rendezvous time and all the objects with an identical position).

Regarding the Generalised Pattern Search algorithm, a 2N-direction complete polling strategy has been considered to characterise the neighbourhood of a point. That is, a positive and negative variation of each of the variables is evaluated and the point that gives the most advantageous value for the objective function is considered in the subsequent iteration.

Regarding the Evolutionary Algorithm, a Genetic Algorithm has been selected. Specifically, its initial population considers 49 random points and one of the initial guesses, and the evaluation is halted when the objective function has not achieved a significant improvement for 50 generations.

Table 6 shows the Δt ratio between the worst aggregated value for the optimal rendezvous time (corresponding to the optimality upper bound) and the worst aggregated value for the feasible rendezvous time (corresponding to the feasibility upper bound). The values reported for the Δt ratio are in general close to 1, highlighting the limited improvement in rendezvous time and, accordingly, in shortening of the total mission time, which can be achieved by optimizing the controlled variables under the worst case scenario uncontrolled variables. It has to be noted that the problem can be decomposed into each individual constellation plane. Thus, the depicted solutions represent the removal of two or three objects within a single plane, for the cases in which it is the first plane cleared or one of the subsequent ones.

Evidently, the Genetic Algorithm requires a considerably larger number of function evaluations, i.e., resolutions of the sequence selection problem. However, when using the *Identical* initial guess, both optimisation techniques converge to the same solution. The reason for such coincidence is that both techniques have been unable to find a solution with a greater rendezvous time than the one directly achieved by such initial guess. In fact, the ratio $\Delta t_{\mathrm{RC}}(\Delta u = 0)/\max(\Delta t_{\mathrm{RC}}) = 0.8361$ for the considered problem. Hence, it is reasonable to think that the initial guess with identical object positions is very likely to be the global optimum. Consequently, it could be used to circumvent the resolution of this rather complex Bilevel Mixed Integer Nonlinear Programming problem in cases with $\Delta t_{\mathrm{RC}}(\Delta u = 0)/\max(\Delta t_{\mathrm{RC}})$ close to 1.

Table 7 shows the aggregated rendezvous time ratio for the multi-plane problem instances, obtained by assembling the plane-decomposed solutions. Obviously, when there is a single object per plane, the feasibility and optimality upper bounds are identical and so are the worst rendezvous times associated to each of those bounds. Moreover, the larger the number of objects within a single plane, the smaller the computed ratio and, thus, more important the influence of the sequence selection in the optimality bounds.

As previously stated, this upper bound improvement is not necessary to accurately determine the feasibility of the considered problem instances, mainly because the majority of the mission time is spent during the drifting

Table 6
Worst rendezvous time ratio for plane-decomposed cases.

Number of objects	Orbital plane	Optimisation technique	Initial guess	Number of iterations	Function evaluations	Δt_{RC} ratio
2	First	P. Search	Identical	26	147	0.9051
2	First	P. Search	Worst	60	270	0.9051
2	First	Genetic	Identical	51	2450	0.9051
2	First	Genetic	Worst	51	2450	0.9051
2	Rest	P. Search	Identical	24	129	0.9247
2	Rest	P. Search	Worst	52	178	0.8138
2	Rest	Genetic	Identical	51	2450	0.9247
2	Rest	Genetic	Worst	51	2450	0.9247
3	First	P. Search	Identical	32	301	0.8798
3	First	P. Search	Worst	50	242	0.7156
3	First	Genetic	Identical	51	2450	0.8798
3	First	Genetic	Worst	68	3249	0.7986
3	Rest	P. Search	Identical	28	250	0.8967
3	Rest	P. Search	Worst	40	214	0.8714
3	Rest	Genetic	Identical	51	2450	0.8967
3	Rest	Genetic	Worst	79	3766	0.8715

Table 7
Worst rendezvous time ratio for multi-plane cases.

P/N	**1**	**2**	**3**
1	1	0.9051	0.8798
2	-	1	0.9435
3	-	-	1

phases. However, this could have a great impact for practical cases in which a larger number of objects is removed from a single constellation plane and the chaser is directly injected into it. Furthermore, such larger number of objects entails a more complex optimisation problem, therefore emphasising the importance of a good initial guess like the one proposed in this work.

6.2. Mothership case

Table 8 shows the requirements imposed to the mothership case. Its associated performance baseline involves two different scenarios: *Scenario 1* considers nine objects to remove within each of the constellation planes. Each of those planes has associated its own mothership, resulting in a mission involving twelve servicing satellites. It is assumed that the whole mothership set is launched into a single injection orbit with the RAAN of the first constellation plane to be cleared. This way, the remaining motherships will coast in the injection orbit until achieving the RAAN of their associated orbit. Then, they will perform a transfer to rendezvous with one of the objects and, after that, phasing manoeuvres will be carried out to remove the remaining ones. Such phasing manoeuvres involve a transfer to a circular orbit with an altitude of 1195 km, so as to keep a safety distance of 5 km from potential active satellites while coasting in the phasing orbit.

Table 8
Mission requirements for the mothership case.

Parameter	Value
Maximum mission time (years)	2
ΔV budget (km·s^{-1})	1

Scenario 2 also involves removing nine objects with each mothership. However, in this case, those objects are distributed among two adjacent constellation planes. As a result, this scenario only requires six servicing satellites. After clearing its first associated constellation plane, each mothership will use a drifting orbit to transfer to the subsequent one. In particular, the selected drifting orbit is identical to the constellation orbit save for its inclination, which will have a value of 87.67 deg.

The aforementioned mission analysis in Huang et al. (2020) states that, for Scenario 1, every mothership is able to clear its assigned plane while fulfilling the maximum mission time and ΔV constraints. However, for Scenario 2, if the ΔV constraint is fulfilled, the upper bound for the mission time ranges from 2.2 years (for the first mothership) to 3.4 years (for the sixth mothership).

The results of the proposed methodology are compared with such conclusions. Fig. 4 shows the results of computing the feasibility bounds for the (N, P) problem instances of up to nine objects. In particular, this figure depicts the problem instance diagrams for the motherships that serve the planes closer to and farther from the injection orbit, respectively labeled as minimum and maximum drift cases. It has to be noted that the feasibility of both mission time and ΔV requirements is considered in such diagrams, i.e., if one of those constraints is infeasible for an instance, such instance is deemed infeasible. In turn, for an instance to be feasible, both requirements have to be feasible for the whole variable domain. The feasibility showed in Fig. 4 is in line with the conclusions of Huang et al. (2020). However, it has been observed that the infeasibility of removing objects dristributed into two planes is due to violations of the maximum mission time constraint. In contrast, the pre-

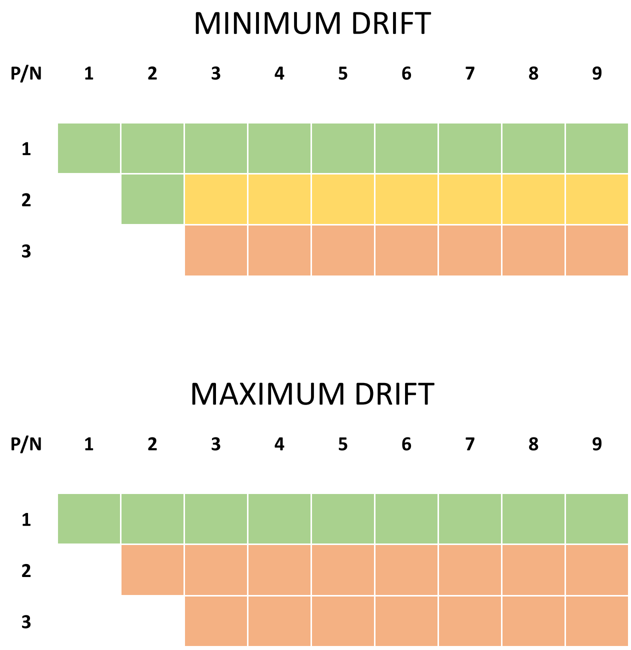

Fig. 4. (N, P) instances diagram for the mothership case.

defined mission parameters result in an unused ΔV of 214.9 m·s^{-1}. Hence, instead of exploring the particular cases in which the removal of objects distributed into two planes is feasible, it would be of great interest to use the techniques explained in Section 5.3 to optimally redistribute such ΔV surplus, thus minimising the mission time.

Fig. 5 depicts the inverse of the summands of Eq. (33), i.e., λ as a function of ΔV for each of the problem variables. It has to be noted that, for a particular constellation plane, the conditions of all of the phasing manoeuvres (save for the first one) are identical. Hence, the ΔV assigned to the phasing of four of the five objects within the first constellation plane is represented by the solid blue line. Likewise, the phasing of three of the four objects within the second plane is characterised by the solid red line. Regarding the

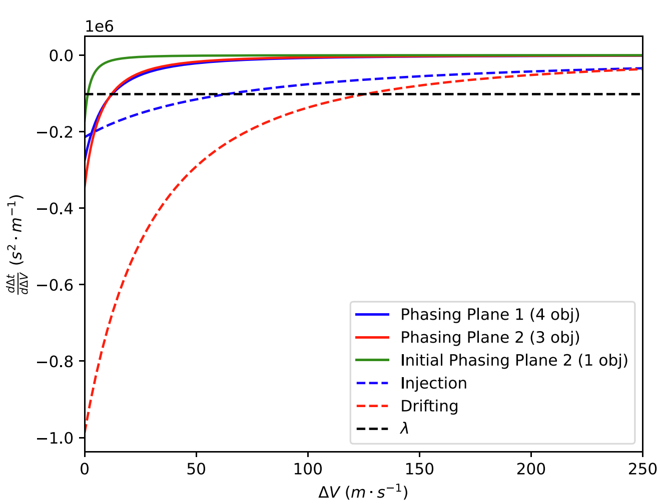

Fig. 5. Mothership case parameter correction.

ΔV assigned to drifting manoeuvres, the dashed blue line labeled as "injection" corresponds to the cost of moving from the injection orbit to the first constellation plane (recall that the injection orbit is also used for drifting in this case), while the "drifting" dashed red line depicts the contribution from drifting between two constellation planes. Furthermore, the value of λ corresponding to a root of Eq. (33) is also portrayed. Consequently, the ΔV assigned to each of the manoeuvres corresponds to the intersection of its corresponding function with the λ value. As the depicted functions show a monotonically increasing behaviour, such intersection is unique and the obtained solution is a global optimum.

Table 9 provides a comparative between the initial inclinations of the injection and drifting orbits, provided by Huang et al. (2020), and the values resulting from the computed λ. Recall that in this scenario, the inclination of the injection orbit is a design parameter. The table also depicts the ΔV allocated to each of those orbits. For the injection case it includes the cost to move from injection orbit to a constellation plane, while for the drifting case it corresponds to transferring to the drifting orbit from one constellation plane, and then back to the next constellation plane. As the drifting phases have a much greater influence in the mission time than the phasing manoeuvres, the bulk of the ΔV surplus is allocated to increase the nodal drift of the injection and drifting orbits.

Table 10 shows the semimajor axes associated to each of the phasing orbits, obtained from the computation of λ, as well as their allocated ΔV. Specifically, phasing orbits 1 to 4 are the ones used to rendezvous with objects 2 to 5 within the first constellation plane. Likewise, phasing orbits 5 to 8 are the ones used to rendezvous with the four objects situated within the second plane. Furthermore, a Corr. and ΔV Corr. stand for the corrections that have to be added to the previous columns so that an integer number of revolutions during the phasing manoeuvres is achieved. Those corrections happen to be considerably small. In addition, Δt Dif. represents the difference in mission time produced by them. The aggregated value of such time difference amounts to 3919 s, which is negligible with respect to the maximum mission time. Therefore, the solution obtained in the phasing and drifting orbit optimisation can be considered as a good approximation of the mission time, regardless of the integrality of the revolutions of the phasing orbits.

Finally, Fig. 6 depicts a comparison between the initial bounds of the mission time and the bounds resulting from the optimised mission, for the problem instance involving nine objects distributed within two constellation planes.

Table 9
Optimised inclination for the injection and drifting orbits.

Orbit	i Ini. (deg)	i Opt. (deg)	ΔV (m·s^{-1})
Injection	86	85.225	63.46
Drifting	87.67	87.174	125.53

Table 10
Optimised phasing orbits.

Phasing orbit	a Opt. (m)	ΔV (m·s^{-1})	a Cor. (m)	ΔV Cor. (m·s^{-1})	Δt Dif. (s)
1	$7.562784 \cdot 10^6$	3.096	$-3.036 \cdot 10^1$	$2.916 \cdot 10^{-2}$	$-2.952 \cdot 10^3$
2	$7.562784 \cdot 10^6$	3.096	$-3.036 \cdot 10^1$	$2.916 \cdot 10^{-2}$	$-2.952 \cdot 10^3$
3	$7.562784 \cdot 10^6$	3.096	$3.678 \cdot 10^1$	$-3.532 \cdot 10^{-2}$	$3.604 \cdot 10^3$
4	$7.562784 \cdot 10^6$	3.096	$3.678 \cdot 10^1$	$-3.532 \cdot 10^{-2}$	$3.604 \cdot 10^3$
5	$7.564505 \cdot 10^6$	1.445	$2.550 \cdot 10^1$	$-2.447 \cdot 10^{-2}$	$2.492 \cdot 10^3$
6	$7.561815 \cdot 10^6$	4.028	$4.150 \cdot 10^{-1}$	$-3.987 \cdot 10^{-4}$	$4.052 \cdot 10^1$
7	$7.561815 \cdot 10^6$	4.028	$4.150 \cdot 10^{-1}$	$-3.987 \cdot 10^{-4}$	$4.052 \cdot 10^1$
8	$7.561815 \cdot 10^6$	4.028	$4.150 \cdot 10^{-1}$	$-3.987 \cdot 10^{-4}$	$4.052 \cdot 10^1$

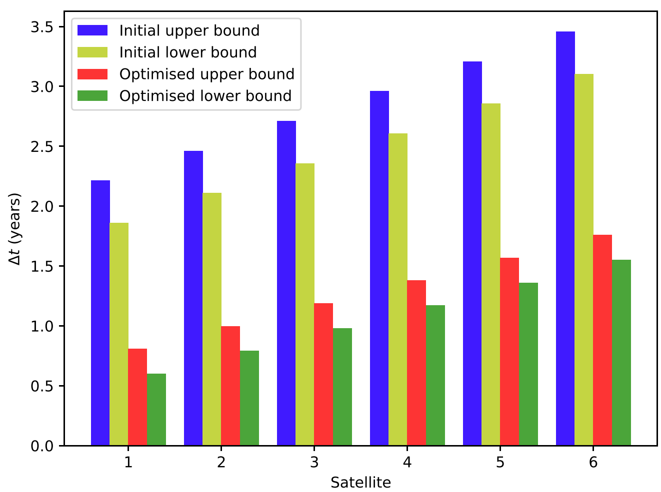

Fig. 6. Mothership case mission time comparison. The optimised cases used the surplus ΔV budget to improve the mission time bounds.

The bounds are obtained by applying the proposed methodology to both the initial solution, that is, the one using the injection, drifting, and phasing orbit definitions from Huang et al. (2020), and the proposed optimized solution that uses the ΔV surplus to reduce mission time. Note that the upper bounds for the initial case coincide with the mission times obtained in Huang et al. (2020), Table 2. Blue and golden bars illustrate, respectively, the initial upper and lower bounds for the initial solution, whereas the red and green bars show, respectively, the upper and lower bounds after the optimisation. Thus, the heights of these bars not only illustrate the reduction in Δt achieved through the proposed modification of injection, drifting and phasing orbits, but also the narrowing between the upper and lower bounds. Note also that the lower and upper bounds are associated with the best- and worst-case scenarios for satellite placement within a respective plane, this being the reason why the difference between the upper and lower bounds is constant across all 6 satellites. The optimised mission shows a remarkable improvement of the mission time, hence fulfilling the maximum mission time constraint with the six servicing satellites and accomplishing the predefined performance baseline for Scenario 2.

In recapitulation, this Section has revisited the mission analysis results of Huang et al. (2020). Taking the two application cases presented therein as a baseline, a performance comparison has been carried out to showcase the capabilities and advantages brought by the proposed methodology. The most remarkable observations and conclusions stemming from this comparison can be summarised as follows:

- Our methodology allows to readily compute the maximum allowable number of objects to be removed with prescribed mission requirements, simply by backwards iteration of Eq. (2), i.e., it provides immediate information on the mission feasibility without the need of performing an actual optimisation. This is exemplified in Section 5.1, where this procedure determines, beforehand, the impossibility of removing more than 3 objects in a single mission with the given constraints.
- For cases in which mission constraints are not met, a quantifiable measure of the non-compliance is obtained (e.g., the amount by which the mission duration is exceeded), along with a clear insight on the situations under which such constraint violation occurs; this gives the mission analyst valuable information for decision-making, thus enabling better informed mission trade-offs, rigorous risk assessments, or revisions of the mission constraints or the overall mission concept. This is also exemplified in Section 5.1, where for Type 1 positioning the maximum mission time is violated by only 2.8 h, as a consequence, deciding that it is not worth to modify the design of the chaser for such a small and improbable violation of the maximum mission time.
- Although the optimality bounds *per se* do not necessarily provide additional information about the problem feasibility, they can nonetheless provide insightful information about the influence of the sequence optimisation for problems with a similar structure; this is, again, exemplified in Section 5.1.
- For scenarios with simultaneous mission constraints to fulfil, the individual feasibility compliance of each of these constraints also provides valuable information to the mission analyst to perform trade-offs between the mission constraints, e.g. exchanging the ΔV budget in benefit of a reduced mission duration; this is illustrated

in Section 5.2 under the Scenario 2, where an unused ΔV of 214.9m s − 1 is identified, and suggested to be optimally redistributed to minimise the mission duration.

7. Conclusions

This manuscript proposes a Constraint Programming framework for the preliminary analysis of space missions. Specifically, it is able to quantify the performance of a set of predefined mission choices with respect to the mission requirements. Moreover, if a poor performance is shown or if the mission choices have not been previously obtained, appropriate mission choices will be generated so that the desired performance is optimised.

This process involves the partitioning of the search space of the concerning problems into problem instances. The feasibility of each of those problem instances with respect to a series of constraints (i.e., the mission requirements) is evaluated. If the feasibility (or infeasibility) of a problem instance is unambiguously determined, a domain pruning process will evaluate the implications of its feasibility for the rest of the problem instances. In turn, if the feasibility of a problem instance is inconclusive, it is partitioned into simpler instances, which will be later evaluated in a similar fashion.

The feasibility of an instance depends on a set of controlled and uncontrolled variables and it is determined by means of bounding the range of constraint values that would be obtained for that set of variables. It has to be noted that such bounds are not unique. Hence, two different sets of bounds have been proposed: in a first case, the feasibility bounds are obtained when using both the controlled and uncontrolled variables to minimise (or maximise) the constraint value, whereas in a second case, the optimality upper bound is obtained when using the controlled variables to minimise the constraint value, while the uncontrolled variables try to maximise it. Consequently, the optimality bounds provide a tighter interval of constraint values, but at the cost of a greater computational complexity. This separates from a strict definition of a Constraint Programming application, where the goal is to identify feasible problems without taking optimality considerations. However, this modification allows to provide a holistic view of the trade-offs involved in mission design and execution.

The proposed methodology has been particularised for two application cases involving constellation-servicing active debris removal missions, namely, a chaser case and a mothership case. The chaser case involves constraints in the fuel consumption and the mission time. Their corresponding feasibility bounds can be readily computed by an algebraic expression. However, obtaining the optimality upper bound requires the resolution of a Bilevel Mixed Integer Nonlinear Programming problem. In such problem, the initial positions of the objects and the removal sequence are simultaneously chosen so as to respectively maximise and minimise the mission time. The upper level is solved using a derivative-free method. Specifically, a Genetic Algorithm and Generalised Pattern Search have been used. Furthermore, an initial guess that provides, under particular circumstances, a near optimal or even the global optimal solution has been figured out. The lower level is modelled as an Integer Linear Programming problem and solved using Branch-and-Bound techniques.

The mothership case involves constraints in the spent ΔV and the mission time. Just like for the chaser case, their corresponding feasibility bounds can be readily computed by an algebraic expression. Nevertheless, the optimality upper bound requires the resolution of a Mixed Integer Nonlinear Programming problem. This problem is divided into two sequential phases. First, the integrality of the revolutions of the phasing orbits is relaxed, obtaining a Nonlinear Programming problem, which is solved by a dual-based method, and then, a correction phase is performed to retrieve the integrality of the revolutions of the phasing orbits.

This methodology has been used to evaluate a preliminary mission analysis of both application cases, developed under ESA's Sunrise project. Regarding the chaser case, it has been determined that its associated mission choices achieve a better performance than the one computed in the preliminary analysis. That is, a more precise knowledge about the performance of the preliminary analysis has been gained, which is an effect of the thorough exploration of the search space performed by the proposed methodology. Regarding the mothership case, it has been shown that the preliminary analysis provides a poor performance. As a result, new values for the semimajor axes of the phasing orbits and the inclinations of the injection and drifting orbits have been computed, thus obtaining significant performance improvements.

In all tested scenarios, the proposed methodology consistently provided a meaningful gain in performance. However, as noted in Section 4.3.2, the performance of the sequence optimization degrades when both the number of objects to be removed $|D|$ and the length of the removal sequence N are large, i.e., $|D|$ and $N \approx |D|$. While this situation does not constitute a problem for the number of objects typical of the application case consider in this work, it can limit the applicability of this methodology to a different mission scenario and would require further model developments to address it.

Although the preliminary mission analysis considered in this work corresponds to two specific scenarios with predefined manoeuvre sequences and constellation characteristics, the flexibility of the proposed methodology allows to extend it to other types of servicing applications. This extension can be considered for future works.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

AB wishes to acknowledge funding from grant PREDOC20-003 of "Universidad Rey Juan Carlos". AB, JLG and HU would also like to acknowledge the Spanish State Research Agency and the European Regional Development Fund for their support through the research grant PID2020-112576GB-C22 (AEI/ERDF, UE). JLG and HU would also like to acknowledge funding grant TED2021-132099B-C32 funded by MCIN/AEI/10.13039/501100011033 and by "European Union NextGenerationEU/PRTR". CC and JLG also acknowledge the European Research Council for the funding received under the European Union's Horizon 2020 research and innovation program within the project COMPASS (grant agreement number 679086). The research also received funding from the European Research Council (ERC) under the European Union's Horizon Europe research and innovation program as part of the GREEN SPECIES project (Grant agreement No 101089265). JLG also aknowledges the funding of his research position by the Italian Ministero dell'Università e della Ricerca, Programma Operativo Nazionale (PON) "Ricerca e Innovazione" 2014–2020, contract RTDA – DM 1062 (REACT-EU).

References

Apt, K., 2003. Principles of Constraint Programming. Cambridge University Press.

Astroscale, 2022. OneWeb, Astroscale, and the UK and European Space Agencies Partner to Launch Space Junk Servicer ELSA-M with 14.8 million Investment. https://astroscale.com/oneweb-astroscale-and-the-uk-and-european-space-agencies-partner-to-launch-space-junk-servicer-elsa-m-with-e14-8-million-investment. Accessed: 12 January 2023.

Barea, A., Urrutxua, H., Cadarso, L., 2020. Large-scale object selection and trajectory planning for multi-target space debris removal missions. Acta Astronaut. 170, 289–301. https://doi.org/10.1016/j.actaastro.2020.01.032.

Barea, A., Urrutxua, H., Cadarso, L., 2022. Dual-based method for global optimization of impulsive orbital maneuvers. J. Astronaut. Sci., 1–26 https://doi.org/10.1007/s40295-022-00357-5.

Barreiro, J., Boyce, M., Do, M., et al., 2012. Europa: A platform for AI planning, scheduling, constraint programming, and optimization. In: 4th International Competition on Knowledge Engineering for Planning and Scheduling (ICKEPS).

Borelli, G., Gaias, G., Colombo, C. et al., 2021. Rendezvous and proximity operations design of an active debris removal service to a large constellation fleet. In: 72nd International Astronautical Congress (IAC 2021), pp. 1–13.

Brettle, H., Lewis, H., Harris, T., et al., 2021. Assessing debris removal services for large constellations. In: 8th European Conference on Space Debris, ESA/ESOC. ESA, pp. 1–8.

Chien, S., Johnston, M., Frank, J. et al., 2012. A generalized timeline representation, services, and interface for automating space mission operations. In: SpaceOps 2012, pp. 1–17.

Colombo, C., Huang, S., Borelli, G., et al., 2021. Mission analysis and design for an active debris removal service for large constellations. In: 8th European Conference on Space Debris, ESA/ESOC. ESA, pp. 1–11.

Federal Communications Commission, 2022. Request for orbital deployment and operating authority for the SpaceX Gen2 NGSO satellite system. Order and Authorization. Released on December 1, 2022. https://docs.fcc.gov/public/attachments/FCC-22-91A1.pdf. Accessed: 12 February 2024.

Dantzig, G., Fulkerson, R., Johnson, S., 1954. Solution of a large-scale traveling-salesman problem. J. Oper. Res. Soc. Am. 2 (4), 393–410.

Forshaw, J., van Steenwijk, R.V., Wokes, S., et al., 2019. Preliminary design of an end-of-life ADR mission for large constellations. Proc. Int. Astronaut. Congr. IAC, 21–25.

Frank, J., Jonsson, A., Morris, R., et al., 2001. Planning and scheduling for fleets of earth observing satellites. International Symposium on Artificial Intelligence, Robotics, Automation and Space.

Gonzalo, J.L., Colombo, C., Di Lizia, P., 2021. Analytical framework for space debris collision avoidance maneuver design. J. Guid., Control, Dynam. 44 (3), 469–487. https://doi.org/10.2514/1.G005398.

Hladik, P.-E., Cambazard, H., Déplanche, A.-M., et al., 2008. Solving a real-time allocation problem with constraint programming. J. Syst. Softw. 81 (1), 132–149.

Inter-Agency Space Debris Coordination Committee, 2021. IADC Space Debris Mitigation Guidelines (iadc-02-01 rev. 3 ed.).

Huang, S., Colombo, C., Gonzalo, J. et al. (2020). Preliminary mission analysis of active debris removal service for large constellations. In:71st International Astronautical Congress (IAC 2020), pp. 1–6.

Jiang, X., Xu, R., 2017. A constraint-programmed planner for deep space exploration problems with table constraints. IEEE Access 5, 17258–17270.

Kessler, D.J., Courpalais, B.G., 1978. Collision frequency of artificial satellites - creation of a debris belt. J. Geophys. Res.-Space Phys. 83 (NA6), 2637–2646. https://doi.org/10.1029/JA083iA06p02637.

Knight, R., Chouinard, C., Jones, G., et al., 2014. Leveraging multiple artificial intelligence techniques to improve the responsiveness in operations planning: Aspen for orbital express. AI Magazine 35 (4), 26–36.

Kuiper Systems LLC (2022). Request for Experimental Authorization. https://apps.fcc.gov/els/GetAtt.html?id=285359. Accessed: 12 January 2023.

Langevin, A., Soumis, F., Desrosiers, J., 1990. Classification of travelling salesman problem formulations. Oper. Res. Lett. 9 (2), 127–132.

Larbi, B., Grzesik, M., Radtke, B., et al., 2017. Active debris removal for mega constellations: Cubesat possible. In: Proceedings of the 9th International Workshop on Satellite Constellations and Formation Flying IWSCFF2017, Boulder, Colorado.

Lewis, H.G., White, A.E., Crowther, R., et al., 2012. Synergy of debris mitigation and removal. Acta Astronaut. 81, 62–68. https://doi.org/10.1016/j.actaastro.2012.06.012.

Lewis, R.M., Torczon, V., 2000. Pattern search methods for linearly constrained minimization. SIAM J. Optim. 10 (3), 917–941.

Liou, J.-C., Johnson, N.L., 2009. A sensitivity study of the effectiveness of active debris removal in LEO. Acta Astronaut. 64 (2–3), 236–243. https://doi.org/10.1016/j.actaastro.2008.07.009.

Miller, C.E., Tucker, A.W., Zemlin, R.A., 1960. Integer programming formulation of traveling salesman problems. J. ACM (JACM) 7 (4), 326–329.

Network Access Associates Ltd (2022). OneWeb webpage. https://oneweb.net/. Accessed: 12 January 2023.

Pemberton, J., 2000. Towards scheduling over-constrained remote sensing satellites. In: Proceedings of the 2d International Workshop on Planning and Scheduling for Space.

Pemberton, J.C., Galiber, F., 2001. A constraint-based approach to satellite scheduling. DIMACS Series in Discr. Math. Theoret. Comput. Sci. 57, 101–114.

Pesant, G., Gendreau, M., Potvin, J.-Y., et al., 1999. On the flexibility of constraint programming models: From single to multiple time windows for the traveling salesman problem. Eur. J. Oper. Res. 117 (2), 253–263.

Rodriguez, J., 2007. A constraint programming model for real-time train scheduling at junctions. Transport. Res. Part B: Methodol. 41 (2), 231–245.

Shaw, P., 1998. Using constraint programming and local search methods to solve vehicle routing problems. In: International conference on principles and practice of constraint programming. Springer, pp. 417–431.

Simonin, G., Artigues, C., Hebrard, E., et al., 2012. Scheduling scientific experiments on the Rosetta/Philae mission. In: International Conference on Principles and Practice of Constraint Programming. Springer, pp. 23–37.

Starlink Services LLC (2022). Starlink webpage. https://www.starlink.com/. Accessed: 12 January 2023.

Wallace, M., 1996. Practical applications of constraint programming. Constraints 1 (1), 139–168.

Weeden, B., 2011. Overview of the legal and policy challenges of orbital debris removal. Space Policy 27 (1), 38–43.