DOI: 10.13009/EUCASS2019-533

DOI: ADD DOINUMBER HERE

Optimal large-scale object selection and trajectory planning for active space debris removal missions

Adrian Barea[⋆†], *Hodei Urrutxua*[⋆] *and Luis Cadarso*[⋆]

[⋆]*European Institute for Aviation Training and Accreditation (EIATA), Rey Juan Carlos University*
Fuenlabrada (Madrid) 28943, Spain
adrian.barea@urjc.es · hodei.urrutxua@urjc.es · luis.cadarso@urjc.es
[†]Corresponding author

Abstract

A branch-and-bound-based algorithm for preliminary design of space debris removal missions has been devised. The proposed algorithm comprises two different levels. Upper level, modelled as an Integer Programming problem, selects a subset of a large pool of candidate objects to be removed so that removed threat value is maximised. Lower level, modelled as a Mixed Integer Nonlinear Programming problem, defines the trajectory that allows to rendezvous with them in an optimal order while a limit mission duration and a Δv budget are imposed as constraints. Performance of this algorithm is demonstrated for a pool of 1000 objects.

1. Introduction

Most objects in low Earth orbit are concentrated in certain privileged orbital regions. A high density of objects in these regions can result in collisions that generate new objects, thus increasing the possibility of subsequent collisions, and potentially leading to a cascade effect that can severely impact future space operations.[6] Further research has concluded that active removal of certain objects is necessary to achieve the stabilisation of the number of resident space objects in the most populated zones.[7,8]

Active removal missions will most likely not target a single object, but instead multiple objects could be deorbited in a single mission. This poses the question of which objects should be removed in a particular mission, and in which order. Branch-and-bound-based techniques have been proposed to solve this problem.[2,4,9,10] However, current approaches in literature decide the objects to be removed prior to the sequence and manoeuvre optimisation or consider a small pool of candidate objects from which a fixed number of objects is selected.

The development of a method that considers a pool of candidate objects that is large enough to be representative of the distribution of the most hazardous objects in LEO region and selects the objects to remove –so that the resulting mission is as impactful as limited amounts of resources make possible– would provide a broader insight for the mission analysis of future active debris removal missions.

This research proposes an innovative framework based on well-known Operational Research methodology that, given a large set of candidate objects with an associated threat value, selects a subset of these objects to be removed, and defines the trajectory that allows to rendezvous with them in an optimal order, so that the threat value of the removed objects is maximised, while a limit mission duration and a Δv budget are imposed as constraints. The proposed algorithm comprises two different levels: the upper level selects the objects to be removed so that threat value is maximised, whereas the lower level checks the feasibility of time and Δv constraints while determining the mission sequence and trajectory.

On the one hand, the upper level is described by an Integer Programming problem, which selects the most promising subset of candidate objects. A novel formulation that avoids the appearance of solutions with subtours during the resolution of this problem has been devised. On the other hand, the lower level is described by a Mixed Integer Non-Linear Programming problem, which is broken down in an Integer Programming master problem and a Nonlinear Programming subproblem using Benders decomposition. The master problem and the subproblem are iteratively solved to find tight upper and lower bounds to the optimal solution.

The efficiency of this algorithm lies in the fact that the no-subtour formulation allows the upper level to effectively select the most promising object subsets while the lower level can check the feasibility of selecting such objects without

DOI: 10.13009/EUCASS2019-533

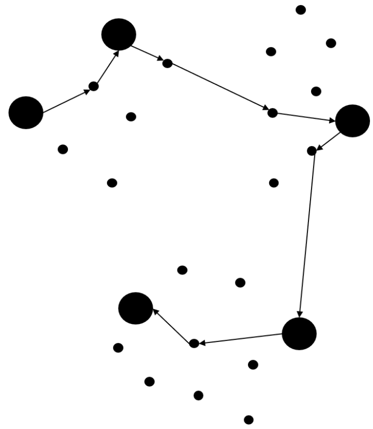

Figure 1: Multi-target debris removal problem.

having to reach the convergence of the upper and lower bounds of the Benders decomposition, which is the most challenging part of the problem because it encapsulates the nonlinearities of the orbital dynamics.

2. Problem Description

Let us consider a pool of candidate spaceborne objects to be potentially removed from orbit. In brief, the problem consists on finding a subset of target objects within the pool, and the sequence in which these should be visited, as depicted in Figure 1, where each spot represents a candidate object and the arrows define the removal sequence of the selected objects. The solution of this problem seeks to maximise a given cost function while fulfilling certain constraints.

Each object in the pool poses a different degree of threat and contributes differently to the risk of operations and sustainability of low-Earth orbit. This level of threat has to be quantified, so that a numerical criterion can be employed to decide the removal of which object offers a bigger reward in terms of an effective decrease of the threat level. Thus, each object is assigned a value, equivalent to a *criticality index*, which accounts for several factors such as their size, mass, lifetime, or how crowded the orbits where they are placed are. In Figure 1 the objects with a higher criticality index are depicted with thicker spots. The aggregated value of the criticality index of all removed objects provides the cost function to be maximised. Indeed, the removed debris has to be as threatening as possible so that the mission has an effective impact in the future evolution of the space environment.[7,8] In the current work, the numerical value of such criticality index, or the criteria for their assignment, are not discussed and are assumed to be known.

The optimisation problem must be carried out making sure that certain constraints are satisfied. The most significant constraint is given by an upper bound of the total fuel required to accomplished the mission (i.e. to visit all the targeted objects), or equivalently, the total Δv. Additionally, the mission time also needs to be bound to decrease operation and design costs; also, the shorter the mission duration, the sooner that threatening objects will be removed, hence reducing the risk of collisions potentially taking place.

Regarding the removal strategy of the selected objects, this study considers that the satellite attaches a de-orbit kit to each of them. This condition is modelled by imposing that the satellite has to co-orbit each of the objects for at least a predefined amount of time, Δt, which stands for the time necessary to perform the installation of the de-orbit kit. Furthermore, it has to be noted that a strategy where the satellite itself transports or shepherds each of the objects to an orbit with a lower orbital lifetime, or to a graveyard orbit, can be implemented by means of simple modifications of the optimisation models. As a result, the framework presented in the following sections can also be used when considering such removal strategies.

As a consequence of the chosen removal strategy, the mission profile comprises two kinds of stages or trajectory arcs, namely co-orbiting arcs and transfer arcs. On the one hand, during a co-orbiting stage, the satellite intercepts a particular object to be removed, installs a de-orbit kit on it, and remains in the same orbit, co-orbiting with the target object (thus following a coasting trajectory arc), until a good opportunity is presented to initiate a transfer (i.e. transfer stage) to rendezvous with the next target object and thus engage a new co-orbiting stage. On the other hand, a transfer stage describes the transition between two co-orbiting stages; specifically, two-impulse transfers have been considered to travel between each pair of adjacent co-orbiting arcs. Thus, co-orbiting and transfer arcs are alternated to obtain the solution trajectory.

In this study, a large number of candidate objects is considered within the pool of objects to be removed (on

DOI: 10.13009/EUCASS2019-533

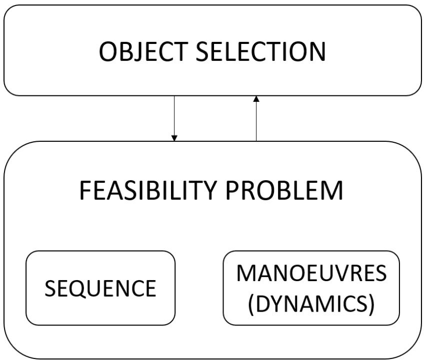

Figure 2: Problem structure.

the order of 1000), so that the problem is meaningful in terms of representing a realistic multi-target mission scenario, where the pool of relevant candidates may easily contain several hundreds or a few thousands of satellites and upper stages. The large number of objects under consideration has strong implications in the combinatorial complexity of the problem.

The other source of complexity lies in the non-linearity of the orbital dynamics. In this regard, it has to be noted that the relevance or criticality of the target objects is not dependent on dynamical aspects, whereas the aforementioned optimisation constraints (fuel and mission duration) clearly depend on orbital dynamics considerations. This realisation provides a good opportunity to formulate a problem with an easily exploitable structure in order to solve it efficiently. That is, the problem can be divided in two parts, namely one that manages the combinatorics of the problem, and another that deals with the orbital dynamics.

As a consequence of that, the proposed problem maximises the reward obtained when removing the selected objects, which is a measurement of the threat that those objects pose to the space environment, while complying with a predefined Δv budget and a maximum mission duration. Then, this problem is divided in two parts, as shown in Figure 2: 1) a *object selection* problem, which optimises the reward of the sequence of objects to be removed; and 2) a *feasibility* problem, which makes use of orbital mechanics to ensure that a mission profile, which complies with prescribed Δv budget and a mission duration constraints, can be generated with the selected removal sequence. This way, the object selection problem passes promising removal sequences through to the feasibility problem, which accepts or rejects them, until the optimal sequence is found.

3. Mathematical Model

This section defines the mathematical formulation of the 'object selection model' and the 'feasibility model' introduced and described in the preceding section.

3.1 Object selection model

The nomenclature used in the formulation of the objective function and constraints of this model is as follows:

- Set D is the set of candidate objects to be visited/removed
- Parameter R_i is the reward obtained when removing object $i \in D$.
- Parameter Δv_{ij} is an estimation of the Δv cost of performing a transfer from object $i \in D$ to object $j \in D$.
- Parameter Δv_T is the total Δv budget for the whole mission.
- Binary variable Y_i is '1' if object $i \in D$ is removed and '0' otherwise.
- Binary variable X_{ij} is '1' if a transfer between objects $i \in D$ and $j \in D$ is performed and '0' otherwise.

This model maximises the sum of the rewards of every object that is visited throughout the mission. Hence, the objective function is given by

$$\max \left\{ \sum_{i \in D} R_i \, Y_i \right\} \tag{1}$$

DOI: 10.13009/EUCASS2019-533

The maximisation procedure needs to ensure that the estimated total Δv spend during the mission does not exceed the Δv_T budget. This can be stated by the following constraint:

$$\sum_{i \in D} \sum_{\substack{j \in D \\ i \neq j}} \Delta v_{ij} X_{ij} \leqslant \Delta v_T \tag{2}$$

An additional constraint is required to ensure that a transfer between two objects is not performed in both directions:

$$X_{ij} + X_{ji} \leqslant 1 \quad \forall i \in D, \forall j \in D : (i \neq j) \tag{3}$$

Also, a transfer has to connect two different objects, i.e. cannot connect an object with itself:

$$X_{ii} = 0 \quad \forall i \in D \tag{4}$$

The orbit where the satellite is initially located, is modelled as the fictitious object $i = 1$, which is automatically removed at the initial time without any Δv cost, i.e. it is assumed that initially the satellite is co-orbiting the fictitious object $i = 1$, namely

$$Y_1 = 1 \tag{5}$$

This particular object has one outgoing transfer and no incoming transfers, as respectively imposed by the following constraints:

$$\sum_{\substack{j \in D \\ j > 1}} X_{1j} = 1 \tag{6}$$

$$X_{i1} = 0 \quad \forall i \in D \tag{7}$$

For the remaining members of the pool of candidate objects contained in the set D, the visited objects have one incoming transfer, i.e.:

$$\sum_{\substack{j \in D \\ i \neq j}} X_{ji} = Y_i \quad \forall i \in D : (i > 1) \tag{8}$$

Conversely, objects which have not been visited, cannot have any outgoing transfer. Consequently, if an object is visited, it has exactly one outgoing transfer, unless it is the last object in the sequence, in which case it has no outgoing transfer either. This constraint implies the following condition:

$$\sum_{\substack{j \in D \\ i \neq j}} X_{ij} \leqslant Y_i \quad \forall i \in D : (i > 1) \tag{9}$$

3.2 Feasibility model

In the formulation of the feasibility model, the following sets are defined:

- D' is a subset of D, which comprises the objects to be removed. This subset is provided by the object selection problem, and is indexed by d.
- A is the set defining the object removal sequence, i.e. it contains the ordering of the objects to be removed, ordered from '1' to $|A|$. It is indexed by α.

As mentioned before, the mission profile comprises two kinds of stages or orbital arcs, namely co-orbiting arcs and transfer arcs. Accordingly, each object $\alpha \in$ A has and associated co-orbiting arc that describes the satellite's trajectory while co-orbiting the object α, and a transfer arc that describes the outgoing trajectory, which begins when the satellite leaves the orbit of the object α and ends when the object $\alpha + 1$ is intercepted. Impulsive manoeuvres are performed in the intersection point of each pair of arcs. It has to be noted that the object $|A|$ only has onse single co-orbiting arc, since the object $|A| + 1$ does not exist. This mission profile is depicted in Figure 3, where the following additional sets are used to define its geometry:

- O is the set that identifies the kind of arc being performed. It takes the value '1' when the satellite is co-orbiting the object α, and '2' for the transfer arc joining the object α and $\alpha + 1$. It is indexed by o.
- Π is the set that identifies the intersection points of the arc being performed with its adjacent arcs, that is, the points where impulsive manoeuvres are performed. It takes the value '1' for the initial point of the arc, and '2' for its final point. It is indexed by π.

DOI: 10.13009/EUCASS2019-533

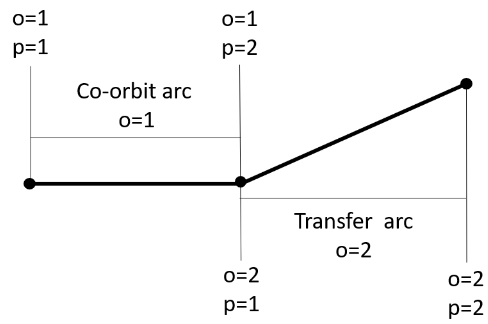

Figure 3: Orbit arcs associated to object $\alpha \in A$.

In the formulation of the feasibility model, the following parameters are defined:

- μ is the gravitational parameter of the Earth.
- $R_\oplus$ is the equatorial radius of the Earth.
- J_2 is the spherical harmonic of degree 2 and order 0 of the Earth's gravity field.
- Δt is the minimum co-orbiting time, i.e. the minimum time lapse during which the satellite has to remain in close operations with an object to successfully eliminate it.
- t_{max} is the maximum mission duration.
- t_d^0 is the reference time in which the orbital parameters of the object $d \in D'$ are known.
- a_d^0, e_d^0, i_d^0, Ω_d^0, ω_d^0 and M_d^0 are the classical orbital elements of the object $d \in D'$ at time t_d^0, where M_d^0 stands for the mean anomaly and the other elements have their usual meaning.

In the formulation of the feasibility model, the following variables are defined:

- $S_{d\alpha}$ is a binary variable that takes the value '1' if the object $d \in D'$ is visited in position $\alpha \in A$ of the sequence, and '0' otherwise.
- $a_{\alpha o}$, $e_{\alpha o}$, $i_{\alpha o}$, $n_{\alpha o}$ and $p_{\alpha o}$ are, respectively, the semi-major axis, eccentricity, inclination, mean motion and semilatus rectum of the orbit $o \in O$ associated to object $\alpha \in A$.
- $\Omega_{\alpha o \pi}$, $\omega_{\alpha o \pi}$ and $M_{\alpha o \pi}$ are, respectively the right ascension of the ascending node, argument of perigree and mean anomaly of the point $\pi \in \Pi$ in the orbit $o \in O$ associated to object $\alpha \in A$.
- t_α^{ref} is the reference time in which the osculating orbital elements of the object $\alpha \in A$ are known.
- $t_{\alpha o \pi}$ is the time in which the satellite is at the point $\pi \in \Pi$ in the orbit $o \in O$ associated to object $\alpha \in A$.
- $\vec{r}_{\alpha o \pi}$ is the position vector of the point $\pi \in \Pi$ in the orbit $o \in O$ associated to object $\alpha \in A$.
- $\vec{v}_{\alpha o \pi}$ is the velocity vector of the satellite when it is at the point $\pi \in \Pi$ in the orbit $o \in O$ associated to object $\alpha \in A$.
- Δv_α^{tran} is the magnitude of the $\Delta\vec{v}$ vector of the manoeuvre performed to depart from the orbit where the satellite is co-orbiting the object $\alpha \in A$.
- Δv_α^{ren} is the magnitude of the $\Delta\vec{v}$ vector of the manoeuvre performed when intercepting the object $(\alpha + 1) \in A$ in order to start co-orbiting with it.

This model minimises the total Δv consumed throughout the mission as shown in the following objective function:

$$\min \left\{ \sum_{\substack{\alpha \in A \\ \alpha < |A|}} \Delta v_\alpha^{ren} + \sum_{\substack{\alpha \in A \\ \alpha < |A|}} \Delta v_\alpha^{tran} \right\} \tag{10}$$

DOI: 10.13009/EUCASS2019-533

where the individual Δv spent in each of the manoeuvres is computed as follows

$$\Delta v_\alpha^{ren} = \left|\vec{v}_{(\alpha+1)(1)(1)} - \vec{v}_{\alpha(2)(2)}\right| \qquad \forall \alpha \in A : (\alpha < |A|) \tag{11}$$

$$\Delta v_\alpha^{tran} = \left|\vec{v}_{\alpha(2)(1)} - \vec{v}_{\alpha(1)(2)}\right| \qquad \forall \alpha \in A : (\alpha < |A|) \tag{12}$$

The constraints (19) and (20) impose that the Cartesian position must be conserved before and after each manoeuvre. This requires the ability to transform Keplerian orbital elements into Cartesian coordinates, which can be achieved by means of the function $\vec{F}$, defined as

$$\begin{bmatrix} \vec{r}_{\alpha o \pi} \\ \vec{v}_{\alpha o \pi} \end{bmatrix} = \vec{F}\left(a_{\alpha o}, e_{\alpha o}, \Omega_{\alpha o \pi}, \omega_{\alpha o \pi}, M_{\alpha o \pi}\right) \qquad \forall \alpha \in A, \forall o \in O, \forall \pi \in \Pi : ((\alpha < |A|) \vee (o < |O|)) \tag{13}$$

The mean motion and the semilatus rectum can be computed as

$$n_{\alpha o} = \sqrt{\frac{\mu}{a_{\alpha o}^3}} \qquad \forall \alpha \in A, \forall o \in O : ((\alpha < |A|) \vee (o < |O|)) \tag{14}$$

$$p_{\alpha o} = a_{\alpha o}\left(1 - e_{\alpha o}^2\right) \qquad \forall \alpha \in A, \forall o \in O : ((\alpha < |A|) \vee (o < |O|)) \tag{15}$$

The variables $a_{\alpha o}$, $e_{\alpha o}$, and $i_{\alpha o}$ remain constant during each of the orbital arcs; these constant variables for the co-orbiting arcs, as well as the reference time t_α^{ref} necessary to propagate the rest of orbital parameters, are given by:

$$\begin{bmatrix} a_{\alpha(1)} \\ e_{\alpha(1)} \\ i_{\alpha(1)} \\ t_\alpha^{ref} \end{bmatrix} = \sum_{d \in D'} S_{d\alpha} \begin{bmatrix} a_d^0 \\ e_d^0 \\ i_d^0 \\ t_d^0 \end{bmatrix} \qquad \forall \alpha \in A \tag{16}$$

The variables $\Omega_{\alpha(1)\pi}$, $\omega_{\alpha(1)\pi}$ and $M_{\alpha(1)\pi}$ can be computed for the co-orbiting arcs at any time $t_{\alpha(1)\pi}$, once their values are given at reference time t_α^{ref}, by means of the following expressions:

$$\begin{aligned} \begin{bmatrix} \Omega_{\alpha(1)\pi} \\ \omega_{\alpha(1)\pi} \\ M_{\alpha(1)\pi} \end{bmatrix} &= \sum_{d \in D'} S_{d\alpha} \begin{bmatrix} \Omega_d^0 \\ \omega_d^0 \\ M_d^0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ n_{\alpha(1)}\left(t_{\alpha(1)\pi} - t_\alpha^{ref}\right) \end{bmatrix} + \\ &+ \frac{3 n_{\alpha(1)} R_\oplus^2 J_2}{4 p_{\alpha(1)}^2}\left(t_{\alpha(1)\pi} - t_\alpha^{ref}\right) \begin{bmatrix} -2\cos\left(i_{\alpha(1)}\right) \\ 4 - 5\sin^2\left(i_{\alpha(1)}\right) \\ \sqrt{1 - e_{\alpha(1)}^2}\left(3\sin^2\left(i_{\alpha(1)}\right) - 2\right) \end{bmatrix} \\ & \qquad \forall \alpha \in A, \forall \pi \in \Pi \end{aligned} \tag{17}$$

However, there are no reference values of the orbital elements for transfer arcs; as a result, the values of $\Omega_{\alpha(2)\pi}$, $\omega_{\alpha(2)\pi}$ and $M_{\alpha(2)\pi}$ between the beginning and the end of every transfer orbit arc are related as follows:

$$\begin{aligned} \begin{bmatrix} \Omega_{\alpha(2)(2)} \\ \omega_{\alpha(2)(2)} \\ M_{\alpha(2)(2)} \end{bmatrix} &= \begin{bmatrix} \Omega_{\alpha(2)(1)} \\ \omega_{\alpha(2)(1)} \\ M_{\alpha(2)(1)} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ n_{\alpha(2)}\left(t_{\alpha(2)(2)} - t_{\alpha(2)(1)}\right) \end{bmatrix} + \\ &+ \frac{3 n_{\alpha(2)} R_\oplus^2 J_2}{4 p_{\alpha(2)}^2}\left(t_{\alpha(2)(2)} - t_{\alpha(2)(1)}\right) \begin{bmatrix} -2\cos\left(i_{\alpha(2)}\right) \\ 4 - 5\sin^2\left(i_{\alpha(2)}\right) \\ \sqrt{1 - e_{\alpha(2)}^2}\left(3\sin^2\left(i_{\alpha(2)}\right) - 2\right) \end{bmatrix} \\ & \qquad \forall \alpha \in A : (\alpha < |A|) \end{aligned} \tag{18}$$

Equations (16) to (18) define the analytical propagation of Keplerian orbital elements.

Additionally, the continuity of the satellite position in the intersection points of the different trajectory arcs needs to be ensured:

$$\vec{r}_{\alpha(1)(2)} = \vec{r}_{\alpha(2)(1)} \qquad \forall \alpha \in A : (\alpha < |A|) \tag{19}$$

$$\vec{r}_{\alpha(2)(2)} = \vec{r}_{(\alpha+1)(1)(1)} \qquad \forall \alpha \in A : (\alpha < |A|) \tag{20}$$

DOI: 10.13009/EUCASS2019-533

The same continuity condition applies to the physical time. Also, the initial and final times of the mission need to be defined, and it must be ensured that the final time of the mission is less than the maximum allowable mission duration, namely:

$$t_{\alpha(1)(2)} = t_{\alpha(2)(1)} \qquad \forall\alpha \in A : (\alpha < |A|) \tag{21}$$
$$t_{\alpha(2)(2)} = t_{(\alpha+1)(1)(1)} \qquad \forall\alpha \in A : (\alpha < |A|) \tag{22}$$
$$t_{(1)(1)(1)} = t^{ref}_{(1)} \tag{23}$$
$$t_{(|A|)(1)(2)} = t_{(|A|)(1)(1)} + \Delta t \tag{24}$$
$$t_{(|A|)(1)(2)} \leqslant t_{max} \tag{25}$$

Further constraints need to be applied to the physical time. In particular, time variables must follow a logical chronological sequence. Additionally, the satellite needs to be in close operations with the target objects (i.e. in a co-orbiting stage) at least during a prescribed minimum duration Δt. These constraints can be written as:

$$t_{(1)(1)(1)} \leqslant t_{(1)(1)(2)} \tag{26}$$
$$t_{\alpha(1)(1)} + \Delta t \leqslant t_{\alpha(1)(2)} \qquad \forall\alpha \in A : (\alpha > 1) \tag{27}$$
$$t_{\alpha(2)(1)} \leqslant t_{\alpha(2)(2)} \qquad \forall\alpha \in A : (\alpha < |A|) \tag{28}$$

The set A is a rearrangement of the set D'. Therefore, each of the objects within D' matches exactly exactly one of the objects within the set A, i.e.

$$\sum_{d \in D'} S_{d\alpha} = 1 \qquad \forall\alpha \in A \tag{29}$$

The orbit the satellite initially describes is modelled as the fictitious object $d = 1$, which is automatically removed at initial time. As a result, the first object in the set A, $\alpha = 1$, also corresponds to the very same orbit. This is modelled by imposing that

$$S_{(1)(1)} = 1 \tag{30}$$

4. Solution Approach

Once the mathematical formulation of the problem has been described, in this section we proceed to present the solution approach, i.e. the techniques necessary to overcome the difficulties that appear in the implementation and efficient resolution of the mathematical models presented in the preceding section.

On the one hand, the linear formulation proposed for the object selection problem is prone to the appearance of *subtours*, which is a very serious difficulty for the resolution of this model. The usual technique to deal with subtours is first presented in Sec. 4.1.1; right after, a new linear formulation that avoids them is presented in Sec. 4.1.2.

On the other hand, the feasibility problem is a Mixed Integer Nonlinear Programming problem, which is difficult to solve. The benefits of using Benders decomposition to manage this problem are discussed in Sec. 4.2.

4.1 Object selection: Subtour management

Constraints (8) and (9) of the object selection problem impose that, if an object is removed, it has at most one incoming and one outgoing transfer arc. However, this is not enough to correctly model the proposed mission because closed loops disjoint from the main rectilinear path of transfers would also fulfil that condition. Such disjoint loops are usually referred as *subtours* and constitute a very serious difficulty for the problem at hand. Figure 4 illustrates this concept. Representing the Δv_{ij} parameter as the distance between spots, the black arrows and dashed lines form a feasible rectilinear path of transfers. However, the grey arrows are shorter (require less Δv_{ij}) than the dashed lines, and as a result this solution with subtours would always be preferred over the presented rectilinear path.

The reason why the appearance of subtours is so negative is that plane changes are the most Δv intensive manoeuvres and it is expected that objects that provide important rewards R_i are spread across different orbital planes. As a result, the best solutions of this model are potentially the ones that form subtours accross different orbital planes, because they allow to remove objects from different orbital planes without spending the Δv necessary to perform the necessary plane change manoeuvres. This phenomenon produces a great number of invalid solutions that have to be pruned out before achieving the best solution without subtours, which requires a great computational effort.

Two alternatives are proposed to manage this problem. On the one hand, the usual method to deal with subtours is considered in the upcoming subsection. On the other hand, a new linear formulation for the object selection problem is presented in Sec. 4.1.2, which avoids the appearance of subtours.

DOI: 10.13009/EUCASS2019-533

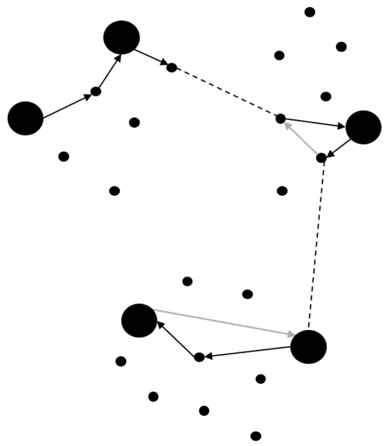

Figure 4: Diagram illustrating the appearance of subtours.

4.1.1 Classical subtour elimination

Current Mixed Integer Linear Programming (MILP) formulations of this kind of models cannot prevent the natural appearance of subtours. The usual approach to deal with them is that, once a feasible solution is found, it gets checked whether it includes any subtours. If it does not, the solution is accepted; if subtours are found, the solution is rejected and a new constraint is added to the problem for each of the subtours found:

$$\sum_{i \in R} \sum_{j \in R} X_{ij} \leqslant |R| - 1 \tag{31}$$

where R is the set of objects that form the subtour. In a subtour, the number of transfers is equal to the number of objects in R. This constraint imposes that the number of transfers among objects in R be, at most, equal to the number of objects minus one. That way, it is not possible that this subtour appears again during the resolution of the problem. Algorithm 1 summarises this subtour elimination strategy.

Algorithm 1 Classical subtour elimination algorithm

```
while solving Object selection problem do
    if feasible solution is found then
        Check subtours
        if Subtours are found then
            Reject solution
            Add subtour elimination constraints
        else
            Solve feasibility problem
        end if
    end if
end while
```

4.1.2 No-subtour formulation

A new formulation that avoids the natural appearance of subtours has been devised. It makes use of a new set $k \in K$ that orders the transfers from '1' to a maximum number of transfers to be made, $|K|$. Then, the discrete variables that represent each of the transfers X_{ij} are changed by new discrete variables X_{ijk} that are '1' when a transfer is made from object $i \in D$ to $j \in D$ in ordinal position $k \in K$, or '0' otherwise. It has to be noted that $\sum_{k \in K} X_{ijk}$ is equivalent to previous variables X_{ij}. This change is introduced in constraints (2) to (4) and (6) to (9).

In addition, it needs to be imposed that for each position $k \in K$ there is at most one transfer:

$$\sum_{i \in D} \sum_{\substack{j \in D \\ i \neq j}} X_{ijk} \leqslant 1 \qquad \forall k \in K \tag{32}$$

DOI: 10.13009/EUCASS2019-533

The appearance of subtours is prevented by means of adding the following constraints. They impose that if there is a transfer in position $(k+1) \in K$, its initial object is the last of the transfers in position $k \in K$. In turn, if there is not a transfer in position $k \in K$, then there is not a transfer in $(k+1) \in K$ either:

$$\sum_{\substack{i\in D\\ i\neq j}} X_{ijk} \geqslant \sum_{\substack{i\in D\\ i\neq j}} X_{ji(k+1)} \qquad \forall j \in D : (j > 1)\,, \quad \forall k \in K : (k < |K|) \tag{33}$$

4.2 Feasibility problem: Benders decomposition

The feasibility problem is a Mixed Integer Nonlinear Programming problem (MINLP), i.e. it is formulated with a nonlinear objective function and constraints while including continuous and discrete variables. Currently, this kind of problems is intrinsically difficult to solve. A possible approach that can be used to manage it is well-known Benders decomposition.

Benders decomposition[1,5] is a technique that can benefit the resolution of certain problems whose structure can be decomposed into two easier problems. This method solves iteratively these two problems in order to achieve the optimal solution of the original problem.

Specifically, this technique decomposes the original problem into a *master problem*, which provides a lower bound of the optimal solution of the original problem, and a *subproblem*, which provides an upper bound of the optimal solution. The solutions of each of these problems provide information to one another, such that the alternating resolutions of each of them provides tighter and tighter bounds of the optimal solution, until their distance is less than a predefined tolerance.

Applying Benders decomposition to the feasibility problem results in a MILP master problem and a Nonlinear Programming (NLP) subproblem, which are more easily tractable than the original MINLP problem.

Moreover, it has to be noted that the purpose of the feasibility problem is not to achieve an optimal solution, but rather to check whether it is possible to configure a mission with the selected object subset, such that it complies with prescribed t_{max} and Δv_T constraints. To take advantage of that, the stop criterion of this specific algorithm is that, regardless of the distance between the upper and lower bounds of the optimal Δv, if the Δv achieved by the master problem is greater than Δv_T, then the object combination is infeasible; conversely, if the Δv achieved by the subproblem is less than Δv_T, then the object combination is feasible. This stop criterion requires fewer iterations than the conventional one, thus notably reducing the computational time. This process is summarised in Algorithm 2, where the set B, indexed by b, keeps track of the iterations of Benders resolution. The following sections describe the subproblem and master problem obtained when applying Benders decomposition to feasibility problem at hand.

4.2.1 Subproblem

The subproblem comprises the objective function of Eq. (10) and the constraints (11) to (28). However, in Eqs. (16) and (17) the integer variables $S_{d\alpha}$ are substituted by continuous variables $S^{\text{rel}}_{d\alpha}$, as follows:

$$\begin{bmatrix} a_{\alpha(1)} \\ e_{\alpha(1)} \\ i_{\alpha(1)} \\ t^{\text{ref}}_{\alpha} \end{bmatrix} = \sum_{d\in D'} \left(S^{\text{rel}}_{d\alpha} \begin{bmatrix} a^0_d \\ e^0_d \\ i^0_d \\ t^0_d \end{bmatrix} \right) \qquad \forall \alpha \in A \tag{34}$$

$$\begin{aligned} \begin{bmatrix} \Omega_{\alpha(1)\pi} \\ \omega_{\alpha(1)\pi} \\ M_{\alpha(1)\pi} \end{bmatrix} &= \sum_{d\in D'} \left(S^{\text{rel}}_{d\alpha} \begin{bmatrix} \Omega^0_d \\ \omega^0_d \\ M^0_d \end{bmatrix} \right) + \begin{bmatrix} 0 \\ 0 \\ n_{\alpha(1)} \left(t_{\alpha(1)\pi} - t^{\text{ref}}_{\alpha} \right) \end{bmatrix} + \\ &+ \frac{3 n_{\alpha(1)} R^2_{\oplus} J_2}{4\, p^2_{\alpha(1)}} \left(t_{\alpha(1)\pi} - t^{\text{ref}}_{\alpha} \right) \begin{bmatrix} -2\cos\left(i_{\alpha(1)}\right) \\ 4 - 5\sin^2\left(i_{\alpha(1)}\right) \\ \sqrt{1 - \left(e_{\alpha(1)}\right)^2} \left(3\sin^2\left(i_{\alpha(1)}\right) - 2\right) \end{bmatrix} \qquad \forall \alpha \in A, \forall \pi \in \Pi \end{aligned} \tag{35}$$

Moreover, a new constraint is introduced in the subproblem. Its purpose is to fix the value of the $S^{\text{rel}}_{d\alpha}$ variables with the values computed in the previous master problem resolution, namely $\widehat{S}_{d\alpha}$, while obtaining the dual variables $\lambda_{d\alpha}$ necessary to compute a new Benders cut for the master problem of the following iteration:

$$S^{\text{rel}}_{d\alpha} = \widehat{S}_{d\alpha} : \lambda_{d\alpha} \qquad \forall d \in D', \quad \forall \alpha \in A \tag{36}$$

DOI: 10.13009/EUCASS2019-533

4.2.2 Master problem

The master problem comprises the constraints (29) and (30) and a new objective function:

$$\min(\theta) \tag{37}$$

Moreover, Benders cuts are introduced to discard suboptimal object sequences. The parameters that configure these cuts are:

- $\widehat{S}^b_{d\alpha}$: Value of $S_{d\alpha}$ variables obtained in the resolution of the master problem in iteration $b \in B$.
- $\widehat{obj}^b$: Value of the subproblem objective function obj obtained in iteration $b \in B$.
- $\widehat{\lambda}^b_{d\alpha}$: Value of dual variables $\lambda_{d\alpha}$ of constraint (36) obtained in the resolution of the subproblem in iteration $b \in B$.

and the argument θ in Eq. (37) is defined as:

$$\theta \geqslant \widehat{obj}^b + \sum_{d \in D'} \sum_{\alpha \in A} \left(S_{d\alpha} - \widehat{S}^b_{d\alpha} \right) \widehat{\lambda}^b_{d\alpha} \qquad \forall b \in B \tag{38}$$

Algorithm 2 Feasibility problem algorithm

while solving Object selection problem **do**
    if feasible solution without subtours is found **then**
        $LB^0 = -\infty$
        $UB^0 = \infty$
        for $b = 1$ to $|B|$, $b+$ **do**
            Solve Master Problem **Obtaining** $S_{d\alpha}, \theta$
            $\widehat{S}_{d\alpha} = S_{d\alpha}$
            $LB^b = \max\left\{LB^{b-1}, \theta\right\}$
            if $LB^b \geqslant \Delta v_T$ **then**
                Reject solution
            end if
            Solve Subproblem **Obtaining** obj
            $UB^b = \min\left\{UB^{b-1}, obj\right\}$
            if $UB^b \leqslant \Delta v_T$ **then**
                Accept solution
            end if
            $\widehat{S}^b_{d\alpha} = S_{d\alpha}$
            $\widehat{\lambda}^b_{d\alpha} = \lambda_{d\alpha}$
            $\widehat{obj}^b = obj$
            Generate Benders cut
        end for
    end if
end while

5. Computational Experiments

This section intends to demonstrate the performance of the previously presented methodology. First, a sample mission is presented with pool of 1000 candidate objects. Second, the Δv_{ij} estimation process necessary for the object selection problem is described for this sample mission. Third, unit tests of the object selection problem are performed without considering the feasibility problem, in order to show the advantage of the no-subtour formulation over the classic subtour elimination strategy. Last, the complete algorithm is tested. The algorithm has been implemented in GAMS 25.1.1 and solved with CONOPT 4.05 and CPLEX 12.8.0 in a PC featuring an Intel Core i5-7500 and 8GB RAM.

DOI: 10.13009/EUCASS2019-533

5.1 Test case mission data

The mission proposed to test this framework considers 1000 candidate objects and a maximum number of 10 objects to be removed. This mission is defined by the following parameters:

- $|D| = 1000$.
- $|K| = 10$.
- $t_{max} = 1$ year.
- $\Delta t = 1$ hour.

The reference orbital parameters associated to each of the 1000 candidate objects correspond to circular LEO orbits. Such orbital parameters, as well as the rewards obtained when removing objects, are randomly generated using a uniform distribution with the following parameters:

- $R_i = U\,(0, 100)$.
- $t_d^0 = 0$ s.
- $a_d^0 = U\,(6578, 8378)$ km.
- $e_d^0 = 0$.
- $i_d^0 = U\,(0, \pi)$.
- $\Omega_d^0 = U\,(0, 2\pi)$.
- $\omega_d^0 = 0$.
- $M_d^0 = U\,(0, 2\pi)$.

Purely Keplerian dynamics is used for this test case. The reason behind this decision is that the feasibility problem is very nonlinear and nonconvex, thus making it highly dependent on a good initial guess. Providing an inaccurate initial guess would make the tests unable to quantify the quality of the algorithm. As good initial guesses for Keplerian dynamics are easier than the corresponding ones for a dynamics that considers the J2 perturbation, this helps to uncouple the measurement of the performance of the proposed algorithm from the selection of the initial guess.

5.2 Δv_{ij} estimation

The object selection problem requires an estimation of the Δv_{ij} consumed in each of the transfers. As the feasibility problem considers two-impulse transfers and the test case comprises only circular orbits, this estimation has been made by means of Hohmann transfers with optimal split plane changes.

This concept is a generalisation of Hohmann transfers in which the initial and final orbits are not coplanar. The velocities of the initial and final orbits, defined by

$$v_i = \sqrt{\frac{\mu}{r_i}}, \qquad v_j = \sqrt{\frac{\mu}{r_j}} \tag{39}$$

and the velocities at apogee and perigee of the transfer orbit, defined by

$$v_i' = \sqrt{\frac{2\mu}{r_i}\frac{r_j}{r_i + r_j}}, \qquad v_j' = \sqrt{\frac{2\mu}{r_j}\frac{r_i}{r_i + r_j}} \tag{40}$$

are the same as the ones in usual Hohmann transfers. However, the plane change required by the transfer is split between the two impulses, resulting in the following Δv_{ij} estimate:

$$\Delta v_{ij} = \sqrt{v_i^2 + v_i'^2 - 2 v_i v_i' cos\,(\phi)} + \sqrt{v_j^2 + v_j'^2 - 2 v_j v_j' cos\,(\theta - \phi)} \tag{41}$$

where θ is the angle between the initial and final orbital planes and ϕ is the plane change performed during first impulse.

The optimal plane change ϕ happens at a stationary point, i.e. $\mathrm{d}\,(\Delta v_{ij})\,/\mathrm{d}\phi = 0$, which results in the following equation that has to be solved numerically:

$$\frac{v_i v_i' sin\,(\phi)}{\sqrt{v_i^2 + v_i'^2 - 2 v_i v_i' cos\,(\phi)}} - \frac{v_j v_j' sin\,(\theta - \phi)}{\sqrt{v_j^2 + v_j'^2 - 2 v_j v_j' cos\,(\theta - \phi)}} = 0 \tag{42}$$

It can be seen that, if $\phi = 0$, the first term of left hand side of Eq. (42) is null and the value of the second term is negative. In turn, if $\phi = \theta$, the second term is null and the first term is positive. This condition has two implications. On the one

DOI: 10.13009/EUCASS2019-533

hand, it makes it possible to use bracketing root-finding algorithms with $\phi = [0, \theta]$ as initial condition. Brent's method[3] has been used to solve this equation because it is considered a fast and robust algorithm that guarantees the convergence of the solution. On the other hand, for this initial condition, the negative value of the function is associated to the lower bound of the bracket and the positive one to the upper bound, so the resulting root represents a stationary point that turns a negative slope into a positive one. That is, this initial condition guarantees the convergence to a minimum.

5.3 Object selection problem unit test

The first step to demonstrate the performance of the proposed framework is to develop a unit test of the object selection problem, that is, without taking into account the feasibility problem. The purpose of this test is to compare the performance of the new, no-subtour formulation versus the classical subtour elimination strategy. The object selection problem deals with the combinatorial complexity resulting from the number of objects in the candidate pool. As a result, this problem has to be solved in an efficient computational time and with a good scalability with the number of objects, so that the complete algorithm can be solved within an reasonable computational time.

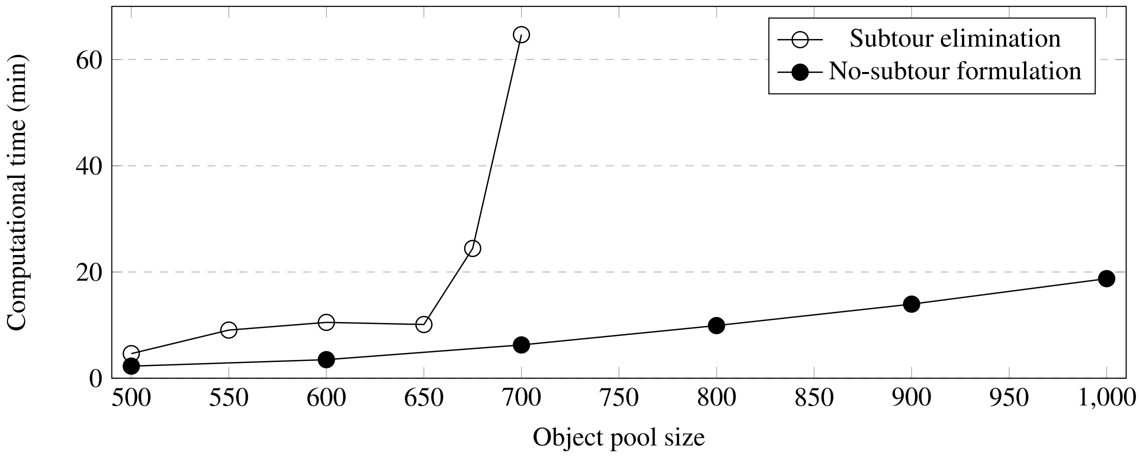

Figure 5: Object selection methods comparison

If a Δv budget of 6 km/s is set for the the test case, Figure 5 shows the computational time spent to solve the object selection problem for a growing pool of candidate objects. The new, no-subtour formulation clearly outperforms the classical subtour elimination strategy and presents a very good scalability with an increasing number of candidate objects in the pool.

The disadvantage of the no-subtour formulation is that it requires a substantially larger number of variables compared to the classical subtour elimination strategy and, as a consequence, exhibits more demanding memory requirements. However, as seen in Table 1, the memory usage of the no-subtour formulation model is affordable for average modern computers.

Table 1: Object selection model comparison

	Subtour elimination	**No-subtour formulation**
Constraints	1003001	1038984
Variables	1001000	10001000
Subtour elimination constraints	$\sim 10^{301}$ [a]	0
Memory usage (Mb)	628	4960

[a]Dynamically generated

In subsequent tests, the no-subtour formulation is used for the resolution of the object selection problem. Figure 6 displays the computational time spent for the resolution of the object selection problem with the complete pool of 1000 objects and an increasing Δv budget. It can be seen that the computational time increases with the available Δv budget, because the larger the Δv budget, the larger the number of possible object combinations and sequences that have to be evaluated.

DOI: 10.13009/EUCASS2019-533

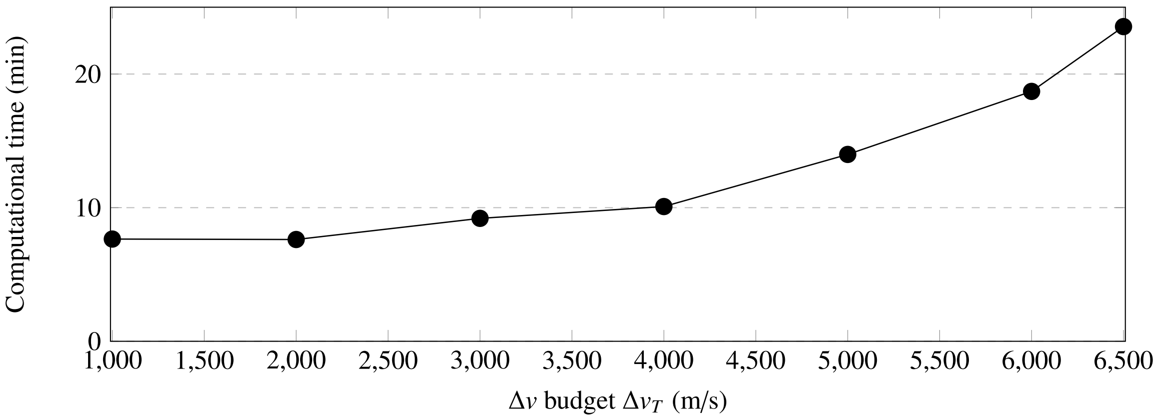

Figure 6: Object selection test

5.4 Complete algorithm test

Now that the no-subtour formulation has shown a very promising scalability with the object pool size, which is a necessary condition for the complete algorithm to be solved in an efficient computational time, the complete algorithm is tested. Figure 7 shows the computational time spent for the resolution of the complete algorithm with the complete pool of 1000 objects and a growing Δv budget. All cases have been solved in under 10 hours, which is a satisfactory result given the complexity of the problem. However, this figure does not behave as Figure 6, where the computational time grows monotonically withthe Δv budget. The reason why this happens is that the computational time necessary to solve this problem is heavily dependent on the data and the search strategy. That is, the faster that high quality solutions are found, the faster the search space is pruned out; however, if numerous mediocre solutions are found, then the search space is pruned out slowly and a large number of feasible solutions have to be explored. This phenomenon is common for all branch-and-bound-based methods, including the object selection test depicted in Figure 6. However, because of the fact that at every time a new feasible solution is found the feasibility problem has to be solved, this effect becomes particularly meaningful.

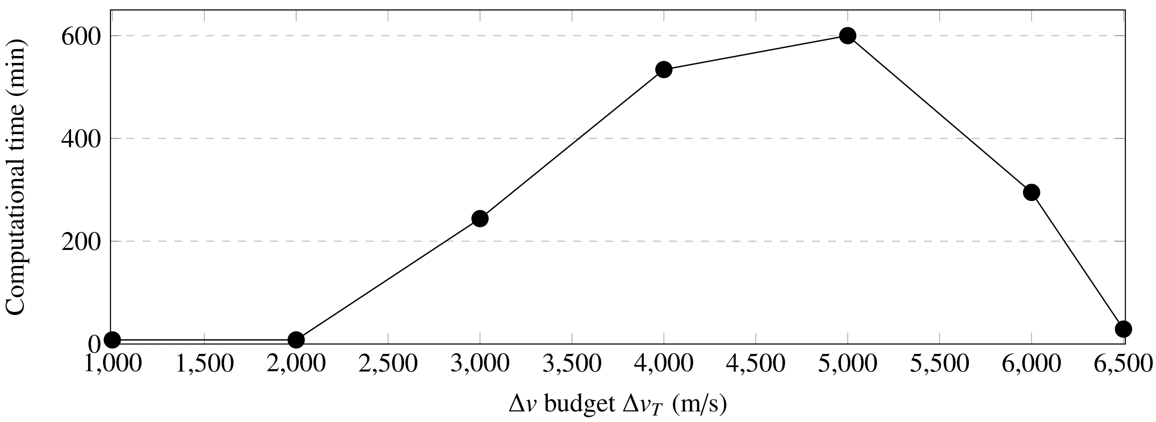

Figure 7: Complete algorithm test

Special attention has to be paid to cases with computational times comparable to the ones achieved by the object selection test, i.e. cases with a Δv budget of 1000, 2000 and 6500 m/s respectively. In order to understand this phenomenon, Table 2 shows the reward obtained in the tests depicted in Figures 6 and 7. It can be seen that the optimal solutions obtained in both tests for the three considered cases are the same. As a result, they represent limit cases of the influence of data and search strategies in the computational time necessary to solve this problem. That is, the optimal solution is easily found and the remaining solutions are efficiently pruned. In fact, their solution could be obtained simply by solving the object selection problem and running the feasibility problem for its optimal solution.

DOI: 10.13009/EUCASS2019-533

Table 2: Reward comparison

Δv budget Δv_T (m/s)	**1000**	**2000**	**3000**	**4000**	**5000**	**6000**	**6500**
Object selection reward	99	241	414	749	870	915	925
Complete algorithm reward	99	241	382	721	866	909	925

6. Conclusions

This work presents a novel branch-and-bound-based algorithm that, given a large set of candidate spaceborne objects with an associated threat value, selects a subset of these objects to be removed, and defines the trajectory that allows to rendezvous with them in an optimal order, so that the aggregated threat value of the removed objects is maximised, while a limit mission duration and a Δv budget are imposed as constraints.

This algorithm comprises two different problems, namely: 1) an object selection problem, described by an MILP problem, that selects the most promising subset of objects so that the aggregated threat value of the removed objects is maximised; and 2) a feasibility problem, described by a MINLP problem, that checks the feasibility of the time and Δv constraints while determining the mission sequence and trajectory.

The natural appearance of subtour in the solutions is a serious difficulty for the object selection problem. A new, no-subtour MILP formulation has been devised, which prevents the appearance of subtours. This novel formulation exhibits a good scalability to handle a large number of candidate objects.

MINLP problems are hard to be treated. That is the reason why the feasibility problem has been broken down into a MILP master problem that determines the removal sequence, and a NLP subproblem that optimises the mission manoeuvres, by using a Benders decomposition. The master problem and the subproblem are iteratively solved until the solution is demonstrated to be either feasible or infeasible, thus not requiring to reach convergence of the upper and lower bounds of the problem.

This methodology has been tested using a 1000 object sample mission. The problem has been solved in satisfactory computational times and a heavy dependence on the search strategy has been observed. This arises the opportunity to further accelerate the problem resolution with a careful selection of such search strategy.

References

[1] J. F. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4(1):238–252, 12/01 1962.

[2] Nicolas Berend and Xavier Olive. Bi-objective optimization of a multiple-target active debris removal mission. *Acta Astronautica*, 122:324–335, MAY-JUN 2016.

[3] R. P. Brent. *An Algorithm with Guaranteed Convergence for Finding a Zero of a Function*, pages 47–58. Algorithms for minimization without derivatives. Prentice-Hall, Englewood Cliffs, NJ, 1973.

[4] M. Cerf. Multiple space debris collecting mission-debris selection and trajectory optimization. *Journal of Optimization Theory and Applications*, 156(3):761–796, MAR 2013.

[5] A. M. Geoffrion. Generalized benders decomposition. *Journal of Optimization Theory and Applications*, 10(4):237–260, 10/01 1972.

[6] DJ Kessler and BG Courpalais. Collision frequency of artificial satellites - creation of a debris belt. *Journal of Geophysical Research-Space Physics*, 83(NA6):2637–2646, 1978.

[7] Hugh G. Lewis, Adam E. White, Richard Crowther, and Hedley Stokes. Synergy of debris mitigation and removal. *Acta Astronautica*, 81:62–68, DEC 2012.

[8] J-C Liou and Nicholas L. Johnson. A sensitivity study of the effectiveness of active debris removal in leo. *Acta Astronautica*, 64(2-3):236–243, JAN-FEB 2009.

[9] Dalal Madakat, Jerome Morio, and Daniel Vanderpooten. Biobjective planning of an active debris removal mission. *Acta Astronautica*, 84:182–188, MAR-APR 2013.

[10] J. T. Olympio and N. Frouvelle. Space debris selection and optimal guidance for removal in the sso with low-thrust propulsion. *Acta Astronautica*, 99:263–275, JUN-JUL 2014.