IAC–15–C1,2,9,x30196

AUTOMATIC PLANNING AND SCHEDULING OF ACTIVE REMOVAL OF NON-OPERATIONAL SATELLITES IN LOW EARTH ORBIT

Juan Manuel Romero Martin
University of Strathclyde, United Kingdom, juan.romero-martin@strath.ac.uk

Marilena Di-Carlo
University of Strathclyde, United Kingdom, marilena.di-carlo@strath.ac.uk

Massimiliano Vasile
University of Strathclyde, United Kingdom, massimiliano.vasile@strath.ac.uk

In this paper two novel strategies to automatically design an optimized mission to de-orbit up to 10 non-cooperative objects per year are proposed, targeting the region within 800 and 1400 km altitude in LEO. The underlying idea is to use a single servicing spacecraft to de-orbit several objects applying two different approaches.The first strategy is analogous to the Traveling Salesman Problem: the servicing spacecraft rendezvous with multiple objects in order to physically attach a de-orbiting kit that performs the re-entry. The second strategy is analogous to the Vehicle Routing Problem: the servicing spacecraft rendezvous with an object, spiral it down to a lower altitude orbit, and spiral up to the next target.
In order to maximize the number of de-orbited non-operative objects with minimum propellant consumption, an optimal sequence of targets is identified using a bio-inspired incremental automatic planning and scheduling discrete optimization algorithm. The incremental planning and scheduling algorithm uses a model based on optimal low-thrust transfer between objects. The optimization of the transfers is realized using a direct method and an analytical propagator based on a first-order solution of the perturbed Keplerian motion. The analytical model takes into account the perturbations deriving from the J2 gravitational effect and the atmospheric drag.

I. INTRODUCTION

Since the beginning of the space era, humankind have put into orbit over 10,000 objects.[1] Only 6% of these are active satellite while the rest are space debris.[1] The growth of space debris population represents a collision threat for satellite and manned spacecraft in Earth orbit. Recent studies have concluded that regions within Low Earth Orbit (LEO) have already reached a critical density of objects which will eventually lead to a cascading process known as the Kessler syndrome.[2] It is expected for the LEO debris population to increase by approximately 30% in the next 200 years.[1,3] The Inter-Agency Space Debris Coordination Committee has issued guidelines to mitigate the growth of space debris.[4] However it has been proved that compliance with these recommendations will not stop the exponential growth. Liou has indeed proved that under the assumption that no spacecrafts are launched after December 2005 the debris population would still grow, driven by collision in the 900-1000 km altitude range.[5] The active removal of five to ten large objects per year is required to stabilize the population.[5] Since in a no-further-release scenario collisions are the only reason for the growth of debris population and since collision probability is a function of the object's cross section, large objects are the main candidate for active removal.[6] Different methods have been proposed for the removal of debris in LEO. These can be contactless method (Ion Beam Shepherd,[7] lasers, solar concentrator[8]) or based on a physical contact with the spacecraft to remove.

This paper focus on the design of a Active Debris Removal (ADR) mission in which a single servicing spacecraft, equipped with an electric engine, removes multiple objects from LEO, in the 800-1400 km altitude region. Each transfer between pairs of objects is optimized in order to be realized with the lowest possible propellant consumption. A bio-inspired discrete decision making algorithm is used to automatic identify the optimal sequence of objects to be removed.

The paper starts with a description of the considered ADR strategies in Section II; the selection method of targets is addressed in Section III. The discrete decision making algorithm for the object sequence selection and the low-thrust transfer model are described in Section IV and V. The obtained results are presented in Section VII,

and some final remarks concludes the paper.

II. Active Debris Removal Strategies

In this paper, two strategies to actively remove objects from LEO are proposed and studied. These two strategies are the De-Orbit Kit Approach and the Spiral Down-&-Up Approach.

In the first strategy, the De-Orbit Kit Approach, the problem is analogous to the Traveling Salesman Problem (TSP). A servicing spacecraft (chaser) rendezvous with multiple objects (targets) in order to physically attach to them a de-orbiting system that performs autonomously the re-entry.

The second strategy, the Spiral Down-&-Up Approach, is analogous to the Vehicle Routing Problem (VRP). A servicing spacecraft rendezvous with an object, grab it and spirals down until a disposal orbit with an altitude of 300 km is reached. Once the disposal orbit is reached, the chaser disengages with the target and spirals up to the next target. The disposal orbit can been seen as the depot of the typical VRP.

Figure 1 illustrates the different mission phases of the two proposed strategies.

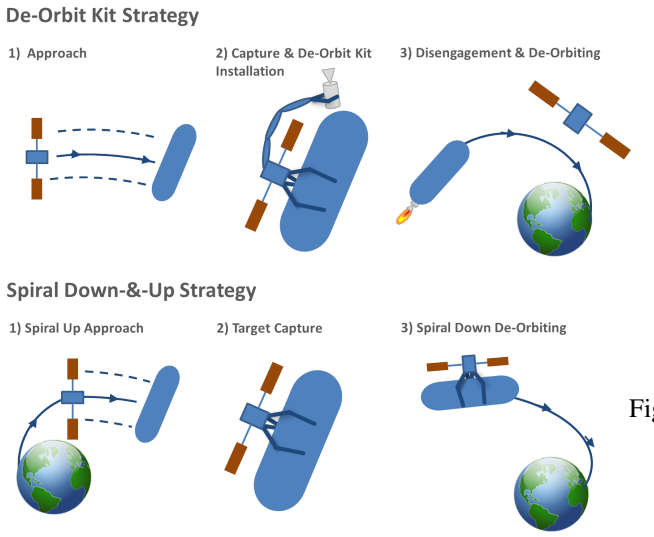

Fig. 1: Mission phases of the two studied ADR Strategies.

III. Target Selection

A catalog of the current objects in LEO is regularly maintained by the North American Aerospace Defence Command (NORAD).[9] Each object in the catalogue is identified by its Two-Line Elements (TLE) set, defining its orbital parameters at a given epoch. For this work, TLE of all objects characterized by perigee altitude $h_p > 800$ km and apogee altitude $h_a < 1400$ km are taken from space-track.org,.[10] In order to target objects more likely to cause collision, only TLE characterized by Radar Cross Section RCS > 1 are considered. The Radar Cross Section is a measure of how detectable is an object with a radar; object with RCS>1 are classified as large. 721 objects characterized by $h_p > 800$ km, $h_a < 1400$ and RCS>1 are found; their distribution in term of semimajor axis vs inclination and semimajor axis vs right ascension Ω is shown in Figure 2.

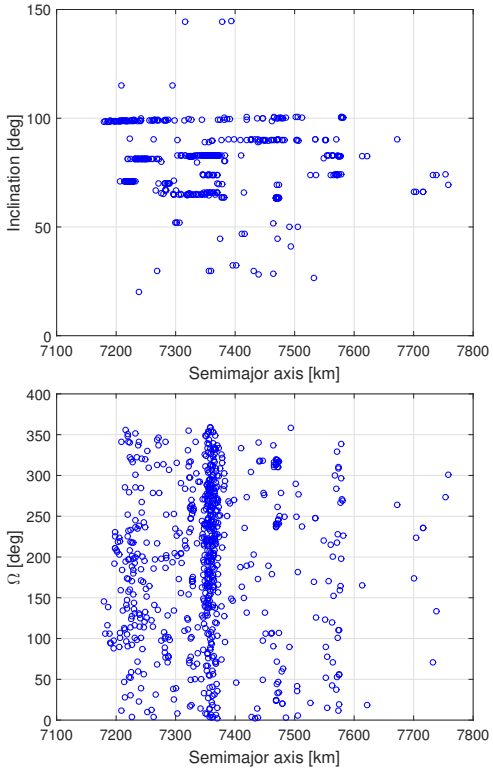

Fig. 2: Semimajor axis, inclination and right ascension of objects in LEO characterized by $h_p > 800$ km, $h_a < 1400$ and RCS>1.

The potential target objects are then further selected based on two main criteria: the right ascension of the ascending node drift due to the second zonal harmonic of the gravity J_2 and the Criticality of Spacecraft Index (CSI).[11] Figure 2 shows that Ω, the Right Ascension of the Ascending Node (RAAN), of the objects is widely spread. Low-thrust maneuvers to change right ascension are particularly expensive and require long time when compared to maneuvers to change other orbital elements. Ruggiero showed that changing 1 degree of right ascension requires 10 days when using optimal thrust angle for the change

of Ω.[12] In this paper the change of Ω is done by taking advantage of the natural rate of nodal regression due to J_2 and its dependence on altitude.[13] The RAAN variation of the servicing spacecraft depends on its altitude and inclination i according to:[14]

$$\dot{\Omega} = -\frac{3}{2} n J_2 \left(\frac{R_\oplus}{a(1-e^2)} \right)^2 \cos i \qquad [1]$$

where n is the orbit's mean motion, $R_\oplus$ the mean Earth's radius and a and e are the orbit's semimajor axis and eccentricity. The effect of a change of semimajor axis on the variation of Ω is greater when the inclination is smaller, because of the $\cos i$ term in the previous Equation. Let us consider a 200 km semimajor axis increase from $a_0 = 7400$ km to $a_1 = 7600$ km. The time spent on the orbit of semimajor axis equal to 7600 km is $T = 10$ days. The resulting difference in Ω with respect to a spacecraft that remains on the orbit of semimajor axis a_0 depends on the inclination through Equation 1

$$\Delta\Omega(a_0, a_1, i, T) = \dot{\Omega}(a_1, i)T - \dot{\Omega}(a_0, i)T \qquad [2]$$

The value of $\Delta\Omega(a_0, a_1, i, T)$ is shown in Figure 3 as a function of the inclination and show that smaller inclination orbits are more favorable for adjustment of right ascension realized by changing the semimajor axis.

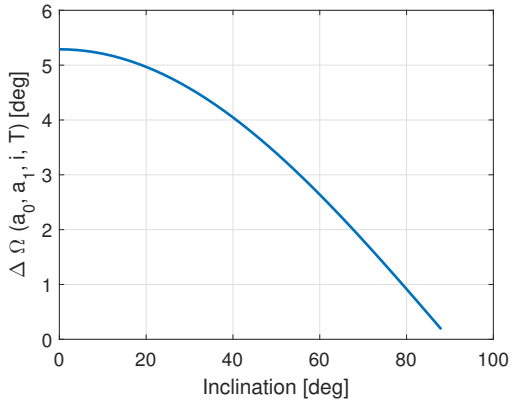

Fig. 3: $\Delta\Omega(a_0, a_1, i, T)$ as a function of the inclination

A further classification of objects with low inclination is realized based on the values of the Criticality of Spacecraft Index (CSI). The Criticality of Spacecraft Index expresses the environmental criticality of objects in Low Earth Orbit taking into account the physical characteristics of a given object, its orbit and the environment where this is located.[11] Figure 4 shows the perigee and the apogee altitudes of the 721 selected objects as a function of the inclination. It can be compared with Figure 8 in Rossi[11] to see that the circled objects in Figure 4 are among the 100 most critical object in term of CSI. These 25 objects are the ones selected for this study. Their orbital elements at epoch 30 May 2015 are reported in Table 5 and shown in Figure 5.

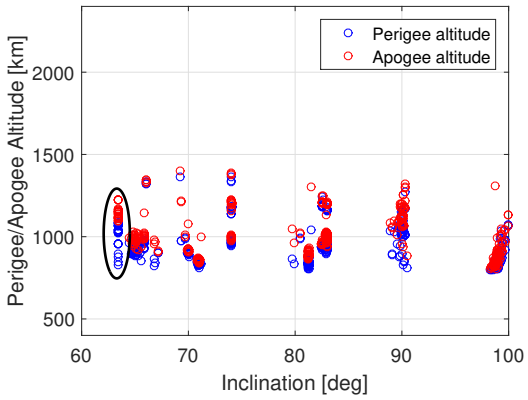

Fig. 4: Perigee and apogee altitude of objects in LEO with $h_p > 800$ km, $h_a < 1400$ and RCS>1

IV. Incremental Planning & Scheduling

The automatic planning & scheduling algorithm used in this paper is based on single objective discrete optimisation algorithm which takes inspiration from the biology of the single cell slime mold Physarum Polycephalum. The Physarum Polycephalum organism has been endowed by nature with a simple but powerful heuristic that can solve complex discrete decision making problems.[15–19] In its main vegetative state, *plasmodium* state, the Physarum Polycephalum forms a network of veins called *pseudopodia*. This network of veins spreads searching for food sources evolving and reshaping with time to find the optimal shape that optimizes the energy required to feed the organism.[19] The flux on the veins varies depending on the distance between the food source and the center of the Physarum. For example, the shortest is the path, the largest is the flux and viceversa.

The Physarum algorithm works modeling the discrete decision making problems into a decision graphs where nodes represent the possible decisions while arcs represent the cost vector associated with decisions. The mechanism of Physarym is analogous to the most commonly known Ant Colony Optimization algorithm.[17] The decision graph is incrementally grown or explored by Virtual Agents using the Physarum-based heuristic. Unlike branch and prune algorithms that use a set of deterministic branching and pruning heuristics, the Physarum algorithm uses probabilistic heuristics to decide to branch

Table 1: List of selected Objects

	NORAD ID	a [km]	e	i [deg]	Ω [deg]	ω [deg]	E [deg]
1	39012	7468.3502	0.0083	63.3824	237.3044	0.8990	359.2169
2	39016	7471.1909	0.0097	63.3825	240.6863	6.5523	353.6732
3	39015	7472.5431	0.0095	63.3828	246.1591	5.6338	354.5722
4	39011	7468.3501	0.0083	63.3835	237.2911	0.9268	359.1897
5	39013	7468.3457	0.0083	63.3851	236.4881	0.7138	359.3978
6	40113	7472.7134	0.0037	63.4023	316.5715	13.7191	346.4819
7	40110	7468.3365	0.0030	63.4026	313.7710	3.4835	356.6392
8	40114	7474.0679	0.0035	63.4027	317.0490	12.1346	348.0508
9	40111	7468.3378	0.0030	63.4036	313.9539	3.3535	356.7684
10	36417	7468.8627	0.0178	63.4045	319.3205	1.2414	136.7021
11	40109	7468.3382	0.0030	63.4048	313.0739	3.0576	357.0626
12	36418	7469.6579	0.0177	63.4050	318.8647	0.8968	90.3057
13	36415	7468.3664	0.0181	63.4057	313.6971	1.5540	358.6008
14	36413	7468.3637	0.0180	63.4064	315.3091	1.5248	358.6278
15	36414	7468.3642	0.0180	63.4079	313.8604	1.2339	358.9091
16	40340	7468.3186	0.0010	63.4084	240.6788	293.4985	66.5005
17	40343	7471.8760	0.0020	63.4093	245.3040	3.4906	356.6256
18	40342	7473.2452	0.0019	63.4096	240.8979	359.1859	0.9133
19	40339	7468.3132	0.0010	63.4097	239.8082	294.9368	65.0721
20	40338	7468.3152	0.0010	63.4108	239.8075	293.4250	66.5729
21	39243	7471.6919	0.0076	63.4150	32.8672	11.6594	348.6165
22	39240	7468.3470	0.0065	63.4154	24.7900	3.7401	356.4085
23	39244	7473.0697	0.0075	63.4156	33.7082	10.7899	349.4691
24	39239	7468.3482	0.0065	63.4158	24.7599	3.6762	356.4729
25	39241	7468.3452	0.0065	63.4170	23.8973	3.3104	356.8339

or prune a vein. To be more specific, branches are never really pruned but the probability of selecting them falls to almost zero. The algorithm has already been extensively tested on a variety of known Travelling Salesman and Vehicle Routing problems with good results.[20–22]

In order to apply the Physarum algorithm, the problem is modeled using a tree-like topology. Starting from a dummy node, that represents the root node, each following children nodes of the root node represent the first servicing task, and its children represent the successive servicing tasks. The decision graph is incrementally grown with time by the virtual agents using the Physarum-based heuristic. Each current node becomes the parent of the following children until an *End Condition* is reached and one full solution is generated. Each arc connecting a parent with a a child has an associated cost evaluated making use of the models presented in Section V.

Both ADR strategies studied in this paper have different *End Conditions* in which a possible full solution condition have been reached. For the De-Orbit Kit strategy, the *End Condition* is reached when the number of installed De-orbit kits reaches a given maximum number $N_{DeObirtKits}$. $N_{DeObirtKits}$ represents the number of kit available on-board of the chaser to be used for the servicing. For the Spiral Down-&-Up strategy, the *End Condition* is reached when the total mission time, $t_{total\_mission}$, has reached a given maximum total mission time, $t_{max\_mission}$.

In this section a brief description of the Physarum's mathematical model is presented, for more detailed description refer to Romero et a. 2014. The mathematical model of Physarum consists mainly in two parts: the decision network exploration and decision network growth. The main parameters of the Physarum solver are summarized in Table 2 and the complete pseudocode is provided in Algorithm 1.

IV.i Decision network exploration

The Decision network exploration is based on the flux through the network of Physarum veins. The flux of the Physarum veins can be modelled as a classical Hagen-Poiseuille flow in cylindrical ducts with variable diameter that varies with time:[17–19]

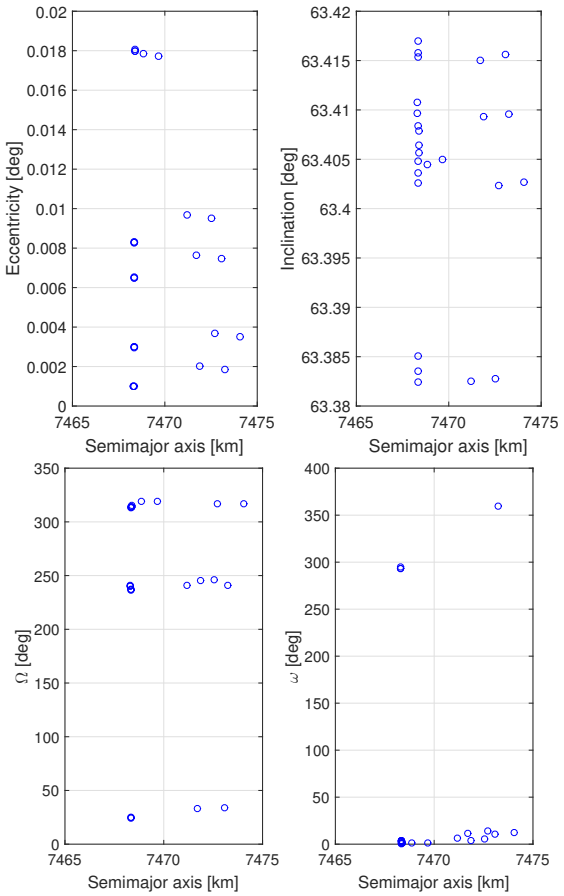

Fig. 5: Orbital elements of the selected objects.

$$Q_{ij} = \frac{\pi r_{ij}^4}{8\mu} \frac{\Delta p_{ij}}{L_{ij}} \qquad [3]$$

where Q_{ij} is the flux between i and j, μ is the dynamic viscosity, r_{ij} is the radius of the vein, L_{ij} is the length of the vein, and Δp_{ij} is the pressure gradient. In this paper, the L_{ij} is substituted by ΔV. For a better understanding of these parameters, they have been illustrated by means of a simple graph in Figure 6.

The variation of the flux through the veins occurs due to the change with time of the radii of the veins. This change are produced mainly by two processes: 1) dilation and 2) contraction of the veins.

The dilation of the veins is caused by the increment of the flowing nutrients through out a vein. The dilation process can be modelled using a monotonic function of the flux:

$$\frac{d}{dt} r_{ij}\Big|_{dilation} = f\left(Q_{ij}\right) \qquad [4]$$

On the other hand, the contraction of the veins is caused by an evaporative effect and it can be modelled as linear function of the radius:

$$\frac{d}{dt} r_{ij}\Big|_{contraction} = -\rho r_{ij} \qquad [5]$$

where $\rho \in [0, 1]$ is a pre-defined evaporation coefficient.

Then, the probability associated with each vein connecting the node i and the node j can be computed using a simple adjacency probability matrix based on fluxes as follow:

$$P_{ij} = \begin{cases} \frac{Q_{ij}}{\sum_{j \in N_i} Q_{ij}} & if\ j \in N_i \\ 0 & if\ j \notin N_i \end{cases} \qquad [6]$$

where N_i is the set of neighbouring veins to a node i.

Table 2: Main Setting parameters for the Physarum solver

m	Linear dilation coefficient, see Eq. (4).
ρ	Evaporation coefficient, see Eq. (5).
GF	Growth factor
N_{agents}	Number of virtual agents.
$N_{Generation}$	Number of Generations.
p_{ram}	Probability of ramification.
r_{ini}	Initial Vein's Radius.
λ	Weight on ramification, see Eq. (7).

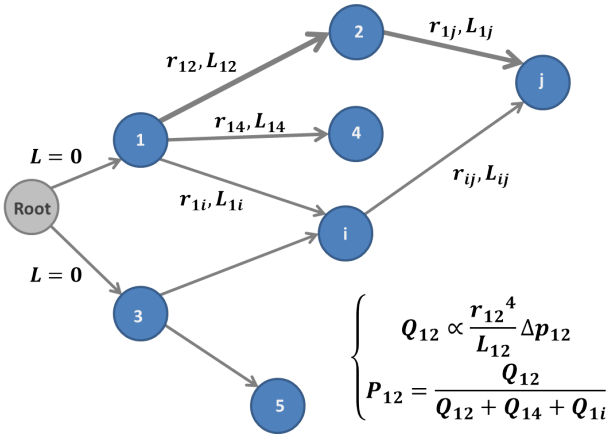

Fig. 6: A simple graph where the thicker arrows represent higher fluxes. In this example $Q_{12} > Q_{14} \rightarrow P_{12} > P_{14}$

IV.ii Growth of the Decision Network

The incremental growth of the decision network is based on a weighted roulette. At every node of the tree, each virtual agent can generate a new branch or move along an existing one. At each node, the virtual agent has a probability p_{ram} of ramification towards new nodes

Algorithm 1 Incremental Physarum Solver

1: initialize m, ρ, GF, N_{agents}, p_{ram}, λ
2: **for** each generation **do**
3: **for** each virtual agent **do**
4: **if** $EndConditoin = true$ **then**
5: Create a new full solution from the current node.
6: continue
7: **end if**
8: **if** $\nu \in \mathcal{U}(0,1) \leq p_{ram}$ **then**
9: using Eq. (7) create a new decision path, building missing links and nodes
10: **else**
11: move on existing graph using Eq. (6).
12: **end if**
13: **end for**
14: contract and dilate veins using Eqs. (4), (5)
15: **if** r_{ij} exceeds upper radius limit **then**
16: block radius increment
17: **end if**
18: update fluxes and probabilities using Eqs. (3), (6)
19: **if** *restart condition* **then**
20: update veins' radii
21: update fluxes and probabilities using Eqs. (3), (6)
22: **end if**
23: **end for**

that are not yet linked with the current one. On line 5 of Algorithm 1, a random number v is drawn from a uniform distribution $U(0,1)$ and the condition $v < p_{ram}$ is verified. Assuming that the agent is at node i, if ramification is the choice, the virtual agent evaluates the set of possible new branches and assigns a probability p_{ij} of constructing a new link from the current node i to a new possible node $j \in \bar{N}_i$, where $\bar{N}_i$ is the set of unlinked nodes (for example nodes 3 and 4 in Figure 7), according to:

$$p_{ij} \propto \frac{1}{L_{ij}^{\lambda}} \qquad [7]$$

where λ is a pre-defined weight exponent. Figure 7 illustrates the concept of possible ramification where dotted lines represent feasible branches not yet existing. If a virtual agent is at node 1, it has a probability p_{ram} of ramification towards the unlinked nodes 3 and 4. If the virtual agent decides to create a new link, a new node is selected according to Equation 7, see line 6 of Algorithm 1.

If a set of linked nodes is available, the virtual agent can decide, with probability $1 - p_{ram}$, to traverse the existing branches in the neighbourhood N_i (see line 8 of Algorithm 1). In the case illustrated in Figure 7 when virtual agent is at node 1, it can explore the already linked nodes 2 or create new links to the unlinked nodes 3 and 4.

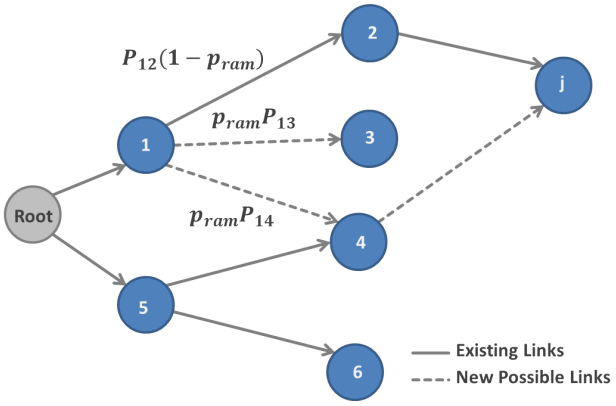

Fig. 7: A simple graph illustrating the ramification towards a new node

IV.iii Problem Transcription

As introduced in Section II, the two proposed ADR strategies are analogous to the typical TSP and VRP. In particular, the De-Orbit Kit strategy is equivalent to the TSP where the goal is to minimize the total distance covered, with the constraint than visiting every town/node once and only once. In this strategy, the goal is to minimize the total ΔV of conducting all the servicing tasks/node only one. In contrast with the regular TSP, where the cost of the arc/link between nodes is constant, in this case, the cost of the arc depends on the the mass of the chaser, and this varies with time due to the reduction of propellant mass, and particularly, due to the reduction of number of de-orbit kits on-board. In order to formulate the problem, other features in addition to the classical TSP ones have been added:

- Only n tasks among S_s are performed. n depends on the number of de-orbit kits available on-board the chaser.
- There is a local duration constrain on the transfers time between tasks, ToF_{min} and ToF_{max}
- There is a minimum waiting time at each task. This waiting time is basically the time required to performed the servicing, $t_{servicing}$

On the other hand, the Spiral Down-&-Up strategy is equivalent to the VRP, where after one service, the vehicle (the chaser in our case) has to return to the depot (the disposal orbit) before proceeding with another service. Similar to the TSP, the goal of the VRP is to minimize the

total distance covered and to conduct every task once and only once. The following features have been added to the traditional VRP:

- There is a global duration constrain on the total mission time, $t_{total\_mission} < t_{max\_mission}$
- There is a local duration constrain on the transfers time between tasks, ToF_{min} and ToF_{max}

As stated previously in this section, the decision graph is incrementally grown by the virtual agents where each node of the of the graph represents a decision. Each of the nodes are connected by arcs, and these arcs have an associated cost, evaluated making use of the model presented in Section V. This cost is the ΔV associated to the transfer between targets. In the Physarum algorithm, the variable L_{ij} in Equations 3 and 7 is replaced by ΔV.

In addition to the waiting time (servicing time) at each node in the TSP, another waiting time has to be added. This additional time is due to the trajectory phasing and is associated with each arc. This waiting time is computed at the same time the arc is evaluated and then added to the departure node. The procedure to compute the phasing waiting time is presented in Section V. The phasing waiting time is not only computed in both the TSP and VRP problem.

V. Low Thrust Transfer Model

In this section the optimization method used to compute the ΔV and time of flight required for the TSP and VRP transfers is described.

V.i Debris Dynamical Model

The debris orbital evolution is generally defined by means of TLE and SGP4 propagator.[23] In this work TLE data are only used to get initial mean elements at epoch t_0 and the state of the objects is propagated considering only J_2 perturbations, since drag is not relevant at the considered altitude.

For each object, therefore, a, e and i are assumed to be constants while Ω and ω change according to:[14]

$$\Omega(t) = \Omega(t_0) - \frac{3}{2}\bar{n}J_2\left(\frac{R_\oplus}{p}\right)^2 \cos i(t - t_0) \quad [8]$$

$$\omega(t) = \omega(t_0) + \frac{3}{2}\bar{n}J_2\left(2 - \frac{5}{2}\sin^2 i\right)^2 (t - t_0) \quad [9]$$

where

$$\bar{n} = n\left[1 + \frac{3}{2}J_2\left(\frac{R_\oplus}{p}\right)^2 \sqrt{1-e^2}\left(1 - \frac{3}{2}\sin^2 i\right)\right] \quad [10]$$

V.ii Time Independence of the Transfers

The rate of change of Ω and ω due to J_2 is different for each selected object and depend on the orbital elements of the specific objects. In the transfer from object A to object B the servicing spacecraft has to correct Ω and ω by an amount:

$$\Delta\Omega(t_0, ToF) = \Omega_B(t_0 + ToF) - \Omega_A(t_0) \quad [11]$$

$$\Delta\omega(t_0, ToF) = \omega_B(t_0 + ToF) - \omega_A(t_0) \quad [12]$$

Different rate of change of Ω and ω for the two objects D_i and D_{i+1} could result in different values of $\Delta\Omega$ and $\Delta\omega$ when transferring from one object to another at different epoch. If this was to be the case, transfer realized at different starting epoch would be characterized by different ΔV. In this section it is shown that, over an arc of time of two years, $\Delta\Omega$ and $\Delta\omega$ for each combinations of objects changes only by a small amount. Therefore the independence of the transfers on the initial epoch can be assumed.

To show this, the following quantities are computed for each combinations of two objects A and B:

$$\begin{aligned}\Delta\Omega_{AB}(t_0, t_{2y}) = &[\Omega_A(t_0 + T_{2y}) - \Omega_B(t_0 + T_{2y})] \\ &- [\Omega_A(t_0) - \Omega_B(t_0)]\end{aligned} \quad (13)$$

$$\begin{aligned}\Delta\omega_{AB}(t_0, t_{2y}) = &[\omega_A(t_0 + T_{2y}) - \omega_B(t_0 + T_{2y})] \\ &- [\omega_A(t_0) - \omega_B(t_0)]\end{aligned} \quad (14)$$

where t_0 is the considered epoch and $T_{2y} = 2$ years. Results are shown in Figure 8 for all the 600 combinations of transfers resulting from the 25 selected objects. It is possible to see that the variations in $\Delta\Omega$ and $\Delta\omega$ are limited to less than 8 degrees over two years, and can be considered negligible for the resulting ΔV of each transfer. Transfer between objects have therefore been computed considering always t_0 as initial epoch.

V.iii Transfer Model

The transfer are optimized in order to reduce the propellant consumption, or ΔV, required to realize them. The acceleration applied by the low-thrust engine over the

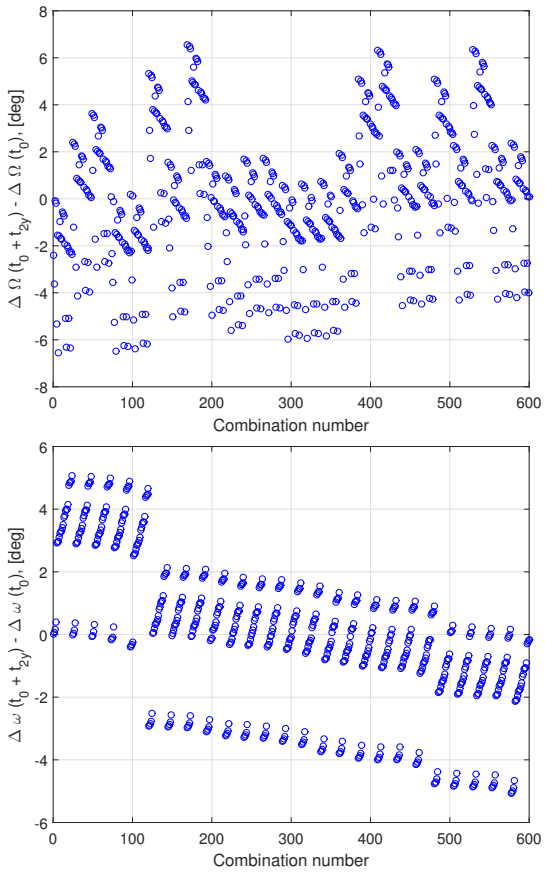

Fig. 8

two thrust arcs can be expressed, in a radial-transverse-normal reference frame as:

$$\begin{cases} a_r &= \varepsilon \cos\beta \cos\alpha \\ a_\theta &= \varepsilon \cos\beta \sin\alpha \\ a_h &= \varepsilon \sin\beta \end{cases} \quad [15]$$

where ε is the acceleration, α is the in-plane azimuth angle and β is the out-of-plane elevation angle.

Each orbit revolution of the servicing spacecraft is divided into four sectors: two thrust arcs (at perigee and apogee) and two coast arcs. In order to consider situations in which thrusting at perigee and apogee could not be the optimal choice, a possible shift of the center of the perigee thrust arc with respect to the perigee itself is considered.

The control variables to optimize are:

- dL_p, amplitude of the perigee thrust arc;
- α_p, azimuth angle of the thrust vector during the perigee thrust arc;
- β_p, elevation angle of the thrust vector during the perigee thrust arc;
- η, angle defining the shift of the first thrust arc (perigee thrust arc) with respect to the perigee;
- dL_a, amplitude of the apogee thrust arc;
- α_a, azimuth angle of the thrust vector during the apogee thrust arc
- β_a, elevation angle of the thrust vector during the apogee thrust arc.

On each thrust arcs, the state of the servicing spacecraft is propagated using an averaged analytical technique.[24] The averaged analytical propagator is based on a set of analytical formulae to propagate the perturbed Keplerian motion over a complete revolution; the averaged orbital elements variations are then numerically propagated. The contribution of the J_2 zonal harmonic and of small perturbations, like the one produced by a low-thrust engine, are included in the analytical formulae. In this work the atmospheric drag has been added to the considered perturbations.

Drag

Zuiani et al. 2015 present an expression for the Gauss' planetary equations in which the variation of the equinoctial elements is expressed as a function of the true longitude $L = \Omega + \omega + \theta$, where θ is the true anomaly.[24] The perturbing accelerations are expressed in a radial transverse normal reference frame. An expression for the drag acceleration in this reference frame is sought. The magnitude of the acceleration acting on a spacecraft due to the atmospheric drag perturbation is:[14]

$$a_D = \frac{1}{2}\rho C_D \frac{A}{m} v^2 \quad [16]$$

where ρ is the atmospheric density, C_D the drag coefficient, A the area of the spacecraft in the direction of the velocity, m the mass of the spacecraft and v its velocity. After a series of manipulations, a_D can be expressed as:

$$a_D = \frac{1}{2}\rho C_D \frac{A}{m}\frac{\mu}{a}\left(\frac{2\Phi(L)}{B^2} - 1\right) \quad [17]$$

where $\Phi(L) = 1 + P_1 \sin L + P_2 \cos L$ and $B = \sqrt{1-e^2}$[24] where $P_1 = e\sin(\Omega+\omega)$ and $P_2 = e\cos(\Omega+\omega)$ are two equinoctial elements.[14] The three components of the acceleration can be expressed in a radial-transverse-normal reference frame, using the previous expression for a_D, as:[24]

$$\begin{cases} a_r = \frac{1}{2}\rho C_D \frac{A}{m}\frac{\mu}{a}\left(\frac{2\Phi(L)}{B^2} - 1\right)\frac{(P_2 \sin L - P_1 \cos L)}{D} \\ a_\theta = \frac{1}{2}\rho C_D \frac{A}{m}\frac{\mu}{a}\left(\frac{2\Phi(L)}{B^2} - 1\right)\frac{(1+P_1 \sin L + P_2 \cos L)}{D} \\ a_h = 0 \end{cases} \quad [18]$$

Substituting this acceleration in the Gauss planetary equations, analytical equations for the variations of the equinoctial elements can be obtained:

$$\begin{cases} a & = a_0 + a_0^2 C_D \frac{A}{m} \left(e_0^2 I_{Drag1} + I_{Drag2} \right) \\ P_1 & = P_{10} + \frac{B_0^2 a_0}{2} C_D \frac{A}{m} \left[\sin\left(\Omega_0 + \omega_0\right) \left(2e I_{Drag4} \right. \right. \\ & \left. \left. + \ 2 I_{Drag6} \right) + 2 \cos\left(\Omega_0 + \omega_0\right) I_{Drag5} \right] \\ P_2 & = P_{10} + \frac{B_0^2 a_0}{2} C_D \frac{A}{m} \left[\cos\left(\Omega_0 + \omega_0\right) \left(2e I_{Drag4} \right. \right. \\ & \left. \left. + \ 2 I_{Drag6} \right) - 2 \sin\left(\Omega_0 + \omega_0\right) I_{Drag5} \right] \\ Q_1 & = Q_{10} \\ Q_2 & = Q_{20} \end{cases} \quad [19]$$

The analytical solutions in Equation V.3 can be superimposed to the one obtained considering J_2 and the low-thrust acceleration.[24] The integrals appearing in the previous equations are hereafter reported:

$$I_{Drag1} = \int \frac{\sin^2\theta \sqrt{1 + e^2 + 2e\cos\theta}}{(1 + e\cos^2\theta)^2} \rho(\theta) d\theta$$

$$I_{Drag2} = \int \sqrt{1 + e^2 + 2e\cos\theta} \rho(\theta) d\theta$$

$$I_{Drag4} = \int \frac{\sqrt{1 + e^2 + 2e\cos\theta}}{(1 + e\cos^2\theta)^2} \rho(\theta) d\theta$$

$$I_{Drag5} = \int \frac{\sin\theta \sqrt{1 + e^2 + 2e\cos\theta}}{(1 + e\cos^2\theta)^2} \rho(\theta) d\theta$$

$$I_{Drag6} = \int \frac{\cos\theta \sqrt{1 + e^2 + 2e\cos\theta}}{(1 + e\cos^2\theta)^2} \rho(\theta) d\theta$$

In the previous expression the density ρ is derived from the exponential atmospheric density model.[14] In order to have analytically integrable expression, ρ is expressed as an expansion in the altitude h with coefficient c computed from a Chebyshev expansion:[25]

$$\rho\left(\theta\right) = \sum_{j=0}^{N} c_j h^j = \sum_{j=0}^{N} c_j \left(\frac{p}{1 + e\cos\theta} - R_\oplus \right)^j \quad [20]$$

V.iv Rendezvous strategy

The rendezvous of the servicing spacecraft with the target object requires the correction of all the six orbital elements. For a given time of flight, ToF, the optimization problems consists in minimizing the ΔV required to realize the transfer subject to the terminal constraints:

$$\mathbf{C} = \begin{bmatrix} a_C - a_T \\ e_C - e_T \\ i_C - i_T \\ \Omega_C(t_0 + ToF) - \Omega_T(t_0 + ToF) \\ \omega_C(t_0 + ToF) - \omega_T(t_0 + ToF) \\ \theta_C(t_0 + ToF) - \theta_T(t_0 + ToF) \end{bmatrix} = \mathbf{0} \quad [21]$$

where t_0 is the departure time and the subscripts C and T denote the chaser servicing spacecraft and target spacecraft, respectively.

In the following the method used to correct a, e, i, Ω and ω will be described. The orbital phasing strategy used to correct θ is described later.

Particular attentions has to be paid to the correction of the right ascension of the ascending node Ω. The preferred method for adjusting it is to take advantage of the natural rate of nodal regression and its dependence on altitude.[13] That is, transferring the spacecraft to lower or higher altitude changes the regression rate relative to the initial orbit and a shift in Ω will build up with time.[13]

Following this approach, the total transfer, characterized by a time of flight ToF, is divided into different phases. In the first one, an optimization problem is solved in order to adjust e, i and ω, in a given time of flight $ToF_{e,i,\omega}$, with the minimum propellant consumption.

The optimization problem of the first phase consist therefore in minimizing $\Delta V_{e,i,\omega}$ subject to the constraints $\mathbf{C}_{e,i,\omega} = 0$ where:

$$\mathbf{C}_{e,i,\omega} = \begin{bmatrix} e_C - e_T \\ i_C - i_T \\ \omega_C(t_0 + ToF_{e,i,\omega}) - \omega_T(t_0 + ToF_{e,i,\omega}) \end{bmatrix} \quad [22]$$

The second phase is realized in a time of flight $ToF_{a,\Omega} = ToF - ToF_{e,i,\omega}$ and its aim is to correct a and Ω, while keeping i and e equal to the target's ones and constraining ω to match the final argument of the perigee of the target orbit at time $t_0 + ToF$. It has to be noted that ω is not expected to change much during the transfer because of the inclination of the selected objects, close to the critical value of 63.43 deg. In order to achieve the final desired a and Ω the following strategy, that takes advantage of the natural nodal regression and its dependence on altitude, is used:[26]

- An optimization problem is solved in order to minimize the ΔV required to move the spacecraft, in a time of flight T_{t1}, from the initial orbit to an appropriate waiting orbit of semimajor axis a_w, while constraining e to be equal to the target's eccentricity e_T. Since this transfer is realized with in-plane thrust

only ($\beta = 0$), no change of inclination will take place and therefore no constraint on i is required.

- The spacecraft remains on the waiting orbit for an appropriate time $T_{w,\Omega}$.
- An optimization problem is solved in order to minimize the ΔV required to move the spacecraft, in a time of flight T_{t2}, from the waiting orbit to the final orbit of semimajor axis a_T, while constraining e and ω to be equal to the target's one, e_T and $\omega_T(t_0 + ToF_{e,i,\omega} + T_{t1} + T_{w,\Omega} + T_{t2})$. Since this transfer is realized with in-plane thrust only ($\beta = 0$), no change of inclination will take place and therefore no constraint on i is required. At the end of this transfer the right ascension of the servicing spacecraft will match the right ascension of the target spacecraft, according to:

$$\begin{aligned}&\Omega_C(t_0 + ToF_{e,i,\omega} + ToF_{a,\Omega}) \\ &= \Omega_T(t_0 + ToF_{e,i,\omega} + ToF_{a,\Omega})\end{aligned} \tag{23}$$

where $ToF_{a,\Omega}$ corresponds to the sum of transfer times and waiting time:

$$T_{t1} + T_{w,\Omega} + T_{t2} = ToF_{a,\Omega} = ToF - ToF_{e,i,\omega} \tag{24}$$

The previous strategy requires therefore the computation of four parameters: T_{t1}, a_w, $T_{w,\Omega}$ and T_{t2}. Four equations are required to solve the problem. The first one derives directly from the available time, Equation 24.

The second equations is derived from Edelbaum theory for the required ΔV for a transfer between circular orbits.[27] This equation can be used because of the small eccentricity of the selected objects. By denoting with $a_{C(e,i,\omega)}$ the semimajor axis of the chaser servicing spacecraft at the end of the first phase, the time to realize the transfer to the waiting orbit can be computed as the ratio between the required ΔV and the spacecraft acceleration:

$$T_{t1} = \frac{k\sqrt{V_{C(e,i,\omega)}^2 + V_w^2 - 2V_{C(e,i,\omega)}V_w}}{\varepsilon} \tag{25}$$

$V_{C(e,i,\omega)}$ is the circular velocity on the orbit of radius $a_{C(e,i,\omega)}$ and V_w is the circular velocity on an orbit of radius a_w. ε is the acceleration of the electric engine and a coefficient $k = 1.5$ is used to multiply ΔV at the numerator in order to allow also for the constraining of e and ω during the transfer. Similarly:

$$T_{t2} = \frac{k\sqrt{V_T^2 + V_w^2 - 2V_T V_w}}{\varepsilon} \tag{26}$$

where now V_T is the velocity on a circular orbit of radius equal to the semimajor axis of the target object.

The last equation used to solve the system of four unknowns is the matching of Ω at the end of the transfer:

$$\begin{aligned}&\Omega_T\left(t_0\right) + \dot{\Omega}_T\left(T_{t1} + T_{t2} + T_{w,\Omega}\right) \\ &= \Omega_C(t_0) + \dot{\Omega_{t1}}T_{t1} + \dot{\Omega}_w T_{w,\Omega} + \dot{\Omega_{t2}}T_{t2}\end{aligned} \tag{27}$$

In the previous equation $\dot{\Omega}_T$ is the drift of right ascension on the target orbit and $\dot{\Omega}_{t1}$ and $\dot{\Omega}_{t2}$ are the drift of right ascension on the transfers to and from the waiting orbit of semimajor axis a_w. $\dot{\Omega}$ changes during these two transfer because of the variation of semimajor axis but it can be approximated to be constants by using a mean semimajor axis a_{t1} for the transfer to the waiting orbit and a mean semimajor axis a_{t2} for the transfer from the waiting orbit to the final target orbit:

$$a_{t1} = \frac{a_{C(e,i,\omega)} + a_w}{2} \tag{28}$$

$$a_{t2} = \frac{a_w + a_T}{2} \tag{29}$$

Equations 24 to 29 allow to compute T_{t1}, a_w, $T_{w,\Omega}$ and T_{t2} required for the change of Ω.

Orbital Phasing

The electric propulsion system provides a limited acceleration, so that each transfer is realized over multiple revolutions. It is hence assumed that by waiting on the initial orbit, and therefore changing the initial true anomaly at departure, $\theta_C(t_0)$, the spacecraft would be able to rendezvous with the target object.[28] This is based on the hypothesis that the low-thrust spiral rigidly rotates and that therefore the difference in true anomaly between the arrival point and the departure point remains constants when the departure time is shifted in time. By denoting this difference with $\Delta\theta$, the equations required to compute the waiting time for orbital phasing, $T_{w,\theta}$ are the following:

$$\begin{aligned}&\theta_T(t_0 + ToF + T_{w,\theta}) - \theta_C(t_0 + T_{w,\theta}) = \\ &\theta_T(t_0 + ToF) - \theta_C(t_0) = \Delta\theta\end{aligned} \tag{30}$$

$$\begin{aligned}&(t_0 + T_{w,\theta}) - t_0 = \sqrt{\frac{a_A^3}{\mu}}\left[2k\pi + E_A(t_0 + T_{w,\theta}) + \right. \\ &\left. - e_A \sin E_A(t_0 + T_{w,\theta}) - E_A(t_0) + e_A \sin E_A(t_0)\right]\end{aligned} \tag{31}$$

$$\begin{aligned}&(t_0 + T_{w,\theta} + ToF) - t_0 = \sqrt{\frac{a_B^3}{\mu}}\left[2k\pi + \right. \\ &E_B(t_0 + T_{w,\theta} + ToF) - e_B \sin E_B(t_0 + T_{w,\theta} + ToF) + \\ &\left. - E_B(t_0) + e_B \sin E_B(t_0)\right]\end{aligned} \tag{32}$$

V.v Problem Transcription

For the strategy in which a deorbiting-kit is attached to the objects to be removed, the servicing spacecraft rendezvous with each object using the strategy described in the previous Section V.4. During the optimization process the drag perturbation is not included in the analytical propagator because the effect of the drag is negligible at the considered altitudes.

For the Spiral Down&Up strategy, the following transfer model is assumed. Considering a situation in which the servicing spacecraft has already realized rendezvous and docking with one of the target object, the total transfer from one object to another consists in the following phases:

- The servicing spacecraft de-orbits the target object by applying a constant tangential thrust, $\alpha = -\pi/2$, over all its orbit, until it reaches a perigee of 300 km altitude. Once this disposal orbit has been reached, the servicing spacecraft disengages with the target.
- The servicing spacecraft increase its semimajor axis by applying a constant positive tangential thrust, $\alpha = \pi/2$, until it reaches the semimajor axis of the next target object
- The servicing spacecraft rendezvous with the next target object using the strategy described in Section V.4

The deorbiting and orbit raising phase are computed using the averaged analytical propagator described in Section V.4 considering both J_2 and drag perturbations.

V.vi Optimization Method

A direct method based on a single-shooting, direct collocation method is used. The MATLAB fmincon-sqp algorithm is used to solve the problem. Four nodes and linear interpolation are used to model the variation of the control variables in the optimization of the first transfer of the rendezvous strategy when e, i and ω are corrected. From four to eight nodes are used for the optimization of the transfer to and from the waiting orbit for the adjustment of Ω.

V.vii Low-Thrust Transfer Surrogate Model

In order to reduce the computational burden in the process of the identification of the optimal sequence of targets, a surrogate model of the low-thrust transfer model is used by the Physarum algorithm to evaluate the cost to link two nodes (Section IV). A surrogate model is a polynomial approximation of the real model and metamodelling has been proven to be an efficient way to imitate the behavior of computationally expensive simulations.

To generate the surrogate model, a limited set of transfers are computed for a limited number of possible masses of the spacecraft and time of flights. The data obtained are then used to construct a surrogate model for the ΔV required for every possible combinations of masses and times. For this study, the Matlab Toolbox DACE (Design and Analysis of Computer Experiment) has been used to construct kriging based metamodels of the low-thrust optimal transfers. As an example, Figure 9 illustrates the surrogate model for the transfer from object 36414 to object 36417.

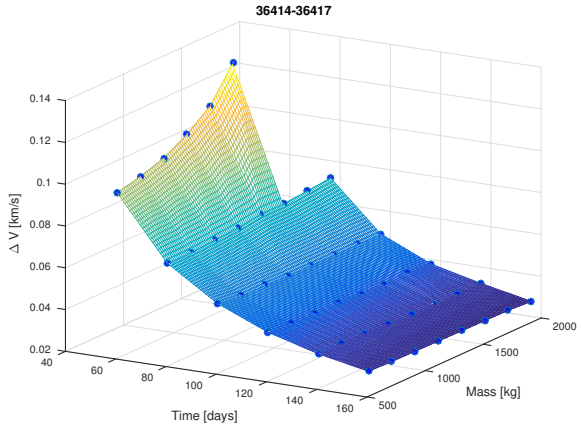

Fig. 9: Surrogate model for the computation of ΔV for the transfer 36414-36417

VI. Mission Definition

For this study, an electric propulsion engine providing 0.1 N of thrust and characterized by a specific impulse of 1600 s I_{sp} is considered. The wet mass of the spacecraft is 1000 kg.

For the De-Orbit kit strategy, 10 de-orbit kits of 100 kg are assumed to be on board the servicing spacecraft, resulting in a total initial mass of 2000 kg. A 100 kg drop is modeled after each transfer to simulate the attachment of the de-orbit kit to the serviced spacecraft. The propellant mass resulting from the transfer is also subtracted from the current mass.

For the Spiral Down-&-Up strategy, each serviced spacecraft is assumed to have a mass of 2000 kg.

VI.i Physarum Algorithm Settings

In addition to the Physarum algorithms parameters m, ρ, GF, N_{agents}, p_{ram}, r_{ini}, $k_{exploration}$ and λ introduced in Section IV, another additional quantities need to be defined. In particular, the set of satellites to be serviced $S_s = \{S_1, S_2, \ldots, S_{N_P}\}$, in our case those presented in

Table 1, the mission start epoch, t_{start}, the maximum mission time, $t_{maximum\_mission}$, the lower and upper boundaries on the time of flight ToF_{min} and ToF_{max} for each leg connecting two servicing tasks i and j, the servicing time for task, $t_{servicing}$ and the maximum allowed change of velocity ΔV_{max}. Another important parameter is the maximum function calls, $Feval_{max}$, where a function call corresponds to the evaluation of an arc.

The values of the parameters of the Physarum algorithm, together with the additional problem parameters used in this study are reported in Table 3.

Table 3: Setting parameters

m	5×10^{-3}
ρ	10^{-4}
GF	5×10^{-3}
N_{agents}	40
$N_{Generation}$	40
p_{ram}	0.7
λ	0
r_{ini}	1
$k_{exploration}$	2
$Feval_{max}$	1×10^5
S_s	All elements in Table 1
t_{start}	February, 22, 2018
$t_{maximum\_mission}$	365 days
ToF_{min}	1 day
ToF_{max}	180 days
$t_{servicing}$	7 days
ΔV_{max}	2.0 km/s

VII. Results

In this section, the results obtained considering the two proposed ADR strategies are presented.

VII.i De-Orbit Kit Results

The sequence of transfer characterized by the lower total time of flight is reported in Table 4. Ten satellites, identified in Table 4 by their NORAD ID, can be serviced in less than one year. m_0 is the initial mass for the transfer and m_f the mass at the end of the transfer. A 100 kg drop in mass is realized after each transfer to account for the attachment of the de-orbiting kit to the serviced satellite. ToF represents the time of flight required to realize the transfer and $T_{w,\theta}$ represents the waiting time on the orbit of the departure object required to obtain the orbital phasing with the arrival satellite. The total time of the mission, considering a servicing time of 7 days for each object, is 358 days.

The variation of mass during the entire servicing time is represented in Figure 10, along with the name of the serviced spacecraft.

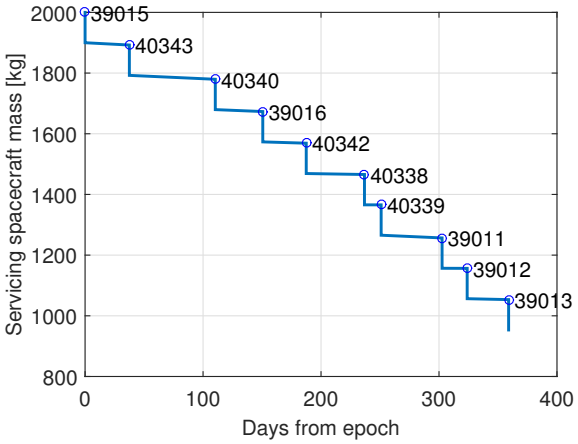

Fig. 10: Mass variation during TSP servicing

The variation of semimajor axis, eccentricity, RAAN and argument of perigee during the transfer from object 40339 to object 39011 is represented in Figure 11 to Figure 14.

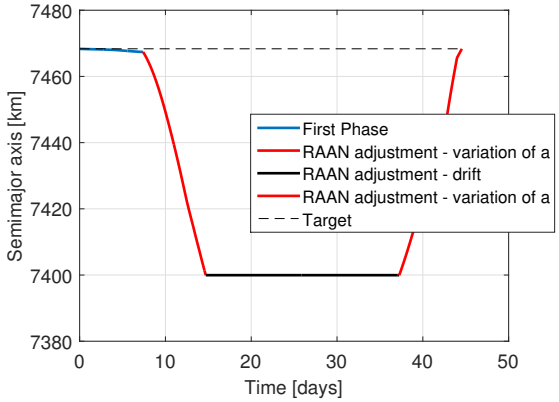

Fig. 11: Semimajor axis variation during transfer 40339-39011.

VII.ii Spiral Down-&-Up Results

The sequence of transfer characterized by the lower total time of flight for the Spiral Down&Up ADR is reported in Table 5, for a total time of flight lower than 1 year. In this Table ToF represent the time required to deorbit the departure object, raise the orbit to the semimajor axis of the next object and then adjust all the other orbital elements.

	Departure Object	Arrival Object	ΔV [km/s]	ToF [days]	$T_{w,\theta}$ [hours]	m_0 [kg]	m_f [kg]
1	39015	40343	0.0628	30.43	2.59	1900.00	1892.40
2	40343	40340	0.1128	65.75	1.78	1792.40	1779.55
3	40340	39016	0.0595	33.14	2.54	1679.55	1673.19
4	39016	40342	0.0429	29.73	2.70	1573.19	1568.89
5	40342	40338	0.0339	42.28	2.06	1468.89	1465.72
6	40338	40339	0.0013	7.05	1.78	1365.72	1365.60
7	40339	39011	0.1116	44.55	2.43	1265.60	1256.63
8	39011	39012	0.0035	14.19	2.07	1156.63	1156.37
9	39012	39013	0.0448	28.04	2.07	1056.37	1053.34
Total	-	-	0.4731	294.17	20.04	-	-

Table 4: Sequence of satellite for Deorbit-Kit ADR strategy.

	Departure Object	Arrival Object	ΔV [km/s]	ToF [days]	$T_{w,\theta}$ [hours]	m_0 [kg]	m_f [kg]
1	39244	36413	1.6307	159.91	2.09	3000.00	2506.11
2	36413	39011	0.9811	182.32	2.41	2506.11	2232.79
Total	-	-	2.6118	373.23	4.5	-	-

Table 5: Sequence of satellite for Spiral Down& Up ADR strategy.

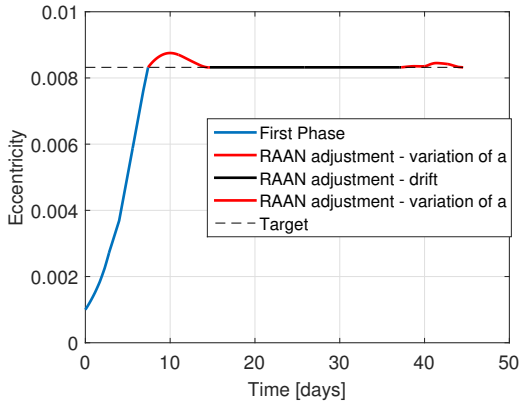

Fig. 12: Eccentricity variation during transfer 40339-39011.

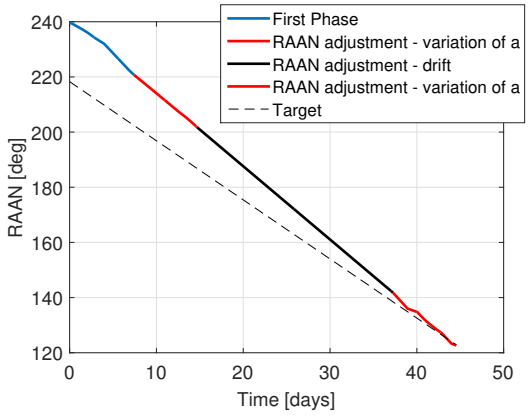

Fig. 13: RAAN variation during transfer 40339-39011.

The deorbit of a spacecraft from the selected altitude region and the subsequent orbit raising to the next target takes a considerable amount of time, making this strategy not feasible for the removal of 5 to 10 objects per year. These results are in agreement with Virgili,[6] who found that 5 objects per year can not be actively de-orbited by grabbing and moving them to a given disposal orbit. Figure 15 shows the variation of perigee altitude of the servicing spacecraft (grabbing object 36413 during the deorbiting phase) and the subsequent orbit raising phase. The total time required is 180.02 days. The shorter orbit raising time is due to the fact that, when the perigee reaches 300 km, the servicing spacecraft dispose of the 2000 kg serviced satellites. The orbit raising phase is therefore realized with a mass of 1000 kg, instead than 3000 kg, resulting in an increased acceleration. Figure 16 shows the variation of RAAN of the servicing spacecraft during the deorbit and orbit raising phases and the variation of RAAN of the next target object, 39011. The RAAN of

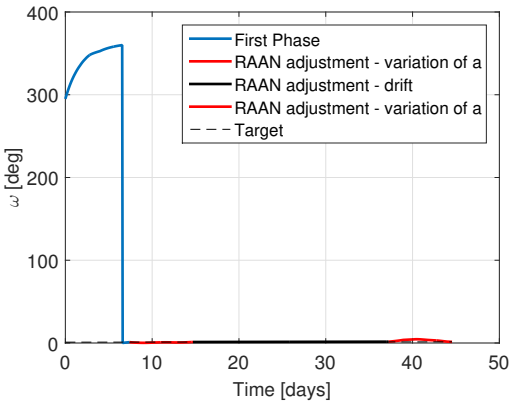

Fig. 14: Perigee argument variation during transfer 40339-39011.

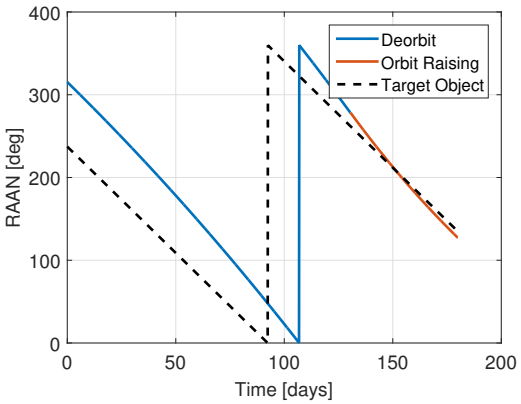

Fig. 16: Variation of Ω of the servicing spacecraft during deorbit of object 36413 and orbit raising to the semi-major axis of object 390111 and variation of Ω of object 39011

the two objects at the end of the orbit raising phase is very close, resulting in a reduced transfer time to the target object.

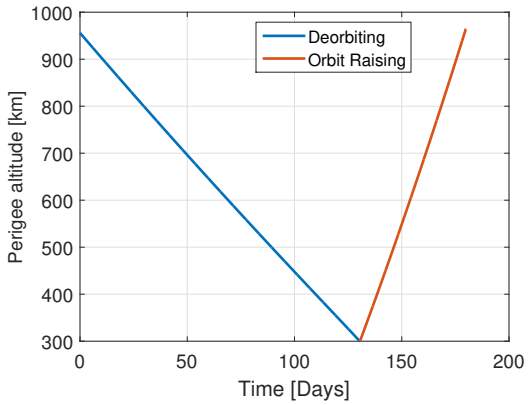

Fig. 15: Variation of the perigee altitude of the servicing spacecraft during deorbit of object 36413 and orbit raising to the semimajor axis of object 39011

VIII. Conclusions

In this paper, two Active Debris Removal strategies have been proposed: De-Orbit Kit and Spiral Down-&-Up. In the Deorbit Kit strategy a servicing spacecraft, equipped with low-thrust propulsion engine, attach a de-orbit device to each objects to be removed; in the Spiral Down-&-Up strategy the servicing spacecraft grab the objects and deorbit them using low-thrust propulsion.

Objects in LEO with altitude in the 800-1400 km range have been considered. The selection of the targets has been conducted assessing, among other factors, their criticality for the space debris environment.

In order to find the optimal sequence of targets to be serviced (with the objective of maximizing the number of de-orbited objects and minimizing the propellant consumption), an innovative incremental planning and scheduling optimization algorithm have been used. To reduce the computational burden, the planning and scheduling algorithm has been used in conjunction with the a use of a surrogate model of the low-thrust transfer model.

Results shows that the De-Orbit Kit strategy is the most effective ADR method. Up to 10 objects per year can be removed with this strategy, meeting the requirements of deorbiting from five to ten objects per year in order to remedy the space debris problem. On the contrary the time required by the Spiral Down-&-Up strategy to remove one object make it an unfeasible solutions for the removal of debris.

References

[1] A. Rossi and G. B. Valsecchi. Collision risk against space debris in Earth orbits. *Celestial Mechanics and Dynamical Astronomy*, 95(1-4):345–356, 2006. ISBN 0923-2958. ISSN 09232958.

[2] D. J. Kessler, N. L. Johnson, J. C. Liou, and M. Matney. The kessler syndrome: implications to future space operations. *Advances in the Astronautical Sciences*, 137(8):2010, 2010.

[3] S. A. Hildreth and A. Arnold. Threats to U.S. National Security Interests in Space: Orbital Debris Mitigation and Removal. 2014.

[4] Inter-Agency Space Debris Coordination Committee (IADC). Inter-agency space debris coordination committee space debris mitigation guidelines. 2007.

[5] J. C. Liou and N. L. Johnson. Instability of the present leo satellite populations. *Advances in Space Research*, 41(7):1046–1053, 2008.

[6] B. B. Virgili and H. Krag. Strategies for active removal in leo. *EMR*, 2(5.0):5–0, 2009.

[7] C. Bombardelli and J. Pelaez. Ion beam shepherd for contactless space debris removal. *Journal of guidance, control, and dynamics*, 34(3):916–920, 2011.

[8] M. Vasile, C. Maddock, and C. Saunders. Orbital debris removal with solar concentrators. In *61st International Astronautical Congress, IAC 2010*, pages Paper–IAC, 2010.

[9] www.norad.mil.

[10] www.space-track.org.

[11] A. Rossi, G. B. Valsecchi, and E. M. Alessi. The criticality of spacecraft index. *Advances in Space Research*, 2015.

[12] A. Ruggiero, P. Pergola, S. Marcuccio, and M. Andrenucci. Low-thrust maneuvers for the efficient correction of orbital elements. In *32nd International Electric Propulsion Conference*, pages 1–13, 2011.

[13] J. E. Pollard. Simplified analysis of low-thrust orbital maneuvers. Technical report, DTIC Document, 2000.

[14] D. A. Vallado and W. D. McClain. *Fundamentals of astrodynamics and applications*. Springer Science & Business Media, 2001.

[15] T. Nakagaki, H. Yamada, and Á. Tóth. Intelligence: Maze-solving by an amoeboid organism. *Nature*, 407(6803):470, 2000.

[16] A. Adamatzky, G. J. Martínez, S. V. Chapa-Vergara, R. Asomoza-Palacio, and C. R. Stephens. Approximating mexican highways with slime mould. *Natural Computing*, 10(3):1195–1214, 2011.

[17] D. S. Hickey and L. A. Noriega. Insights into information processing by the single cell slime mold physarum polycephalum. In *UKACC Control Conference*, pages 2–4, 2008.

[18] A. Tero, R. Kobayashi, and T. Nakagaki. Physarum solver: a biologically inspired method of road-network navigation. *Physica A: Statistical Mechanics and its Applications*, 363(1):115–119, 2006.

[19] R. Kobayashi T. Saigusa T. Nakagaki A. Tero, K. Yumiki. Flow-network adaptation in physarum amoebae. *Theory in Biosciences*, 127(2):98–94, 2008.

[20] L. Masi and M. Vasile. A multidirectional physarum solver for the automated design of space trajectories. July 6-11, 2014.

[21] M. Vasile and V. Becerra. Computational intelligence in aerospace sciences. *Progress in Astronautics and Aeronautics*, 244, 2014.

[22] J. M. Romero Martin, L. Masi, M. Vasile, E. Minisci, R. Epenoy, V. Martinot, and J. Fontdecaba Baig. Incremental planning of multi-gravity assist trajectories. In *65st International Astronautical Congress, IAC 2014*, pages Paper–IAC, 2014.

[23] F. R. Hoots and R. L. Roehrich. Spacetrack report no. 3. *Colorado Springs CO: Air Force Aerospace Defence Command*, pages 1–3, 1980.

[24] F. Zuiani and M. Vasile. Extended analytical formulas for the perturbed keplerian motion under a constant control acceleration. *Celestial Mechanics and Dynamical Astronomy*, 121(3):275–300, 2015.

[25] A. Gil, J. Segura, and N. M. Temme. *Numerical methods for special functions*. Siam, 2007.

[26] B. Chamot and M. Richard. Mission and system architecture design for active removal of rocket bodies in low earth orbit. Master's thesis, Massachusetts Institute of Technology, 2012.

[27] V.A. Chobotov. *Orbital mechanics*. AIAA, 2002.

[28] F. Zuiani and M. Vasile. Preliminary design of debris removal missions by means of simplified models for low-thrust, many-revolution transfers. *International Journal of Aerospace Engineering*, 2012.