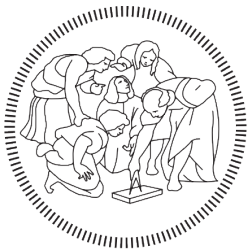


POLITECNICO
MILANO 1863

Department of Aerospace Science and Technologies
Master of Science in Space Engineering

Debris sequence optimization for an active debris removal mission using a multi-indices rating system based algorithm

Author:
Alex Yousef Al Naber
939915

Supervisor:
Prof. Camilla Colombo

Co-Supervisor:
Giacomo Borelli

Academic Year 2022-2023

Dissertation of:	Alex Yousef Al Naber
Supervisors:	Prof. Camilla Colombo
	Dr. Giacomo Borelli

Alex Yousef Al Naber (2023), *Debris sequence optimization for an active debris removal mission using a multi-indices rating system based algorithm*, Master thesis, Politecnico di Milano, Supervisors: Colombo, Camilla and Borelli, Giacomo

Acknowledgement

First of all, I would like to thank my family, who always supported me in my choices and never put pressure on me during my college journey. Then I would like to thank my girlfriend Claudia, who gave me the motivation to finish and without whom I would still be lost. Then I would like to thank my friends Alice, Donya, Anna, Matteo, Marco, Alessandro, Ludovica, Manuel, and Dilson, who always made me laugh during all these years of university, making me perceive them as much less burdensome. Finally, I would like to thank Professor Camilla Colombo and my thesis advisor Giacomo Borelli for their incredible patience with me during the writing of the thesis.

Abstract

In recent years, there has been a rapid growth in the space sector. As it has become more accessible, also thanks to the spread of CubeSats, many agencies, not only public but also private, are gradually populating the Earth's orbits. However, this surely represent a problem due to the overpopulation of main orbits. These orbits are gradually filling up with inactive objects, called space debris, such as nonfunctional and failed satellites, abandoned launch vehicle stages and fragmentation debris from the breakup of derelict rocket bodies and spacecraft .
Moreover, it was study by Donald J. Kessler that the uncontrolled evolution of the space debris population will result in a cascade growth caused by debris collisions. This will pose a great threat to the exploitation of the Earth's orbits by future space assets. In fact, each collision would generate an huge amount of debris, since the high velocity of the satellites and the consequent high energy of the impact, and that would lead to an avalanche effect that would take to a lot more collisions and an exponential growth of the debris.
In order to mitigate this problem it is necessary to remove an adequate number of debris per year and thus counter the exponential growth of the risk of collision. An Active Debris Removal mission aims to locate a target group of debris and deorbit them in a defined order.
With the purpose of selecting an appropriate sequence of debris to deorbit, different evaluation criteria can be used. Most of the literature proposes a bi-objective optimization to solve the problem, aiming to minimize the total ΔV and the total duration of the mission. Other authors focus also on the impact of the debris on the orbital environment. In this dissertation, the function to minimize takes into account the total cost, the debris environmental impact and the debris capture difficulty, through the use of two weighted indices, the environmental index (I_{ENV}) and the operability index (I_{OP}).
This thesis work aim to develop an algorithm that will find the best sequence of debris to deorbit, considering not only the necessary ΔV, but alsothese two indices.
Firstly, the problem will be introduced, talking about the various contributions of the scientific literature on the subject. Then possible mission architectures will be analyzed, highlighting advantages and disadvantages and talking about the architecture selected for this dissertation. It will then be exposed where the debris data used for the dissertation came from, highlighting the distribution of debris in the different orbital belts and explaining how the lack of some data was compensated for. At this point the actual algorithm will be exposed. After discussing the optimization problem to be solved and the different algorithms that can be used, it will be explained how a Branch&Bound algorithm was chosen, which will not only consider the transfer cost but also the two indices to determine the best sequence to deorbit, and all the assumptions made. At the end, the results obtained will be displayed and commented on, and some considerations about future development and possible implementations to the code will be made.

Sommario

Negli ultimi anni si è assistito a una rapida crescita del settore spaziale. Essendo diventato più accessibile, anche grazie alla diffusione dei CubeSat, molte agenzie, non solo pubbliche ma anche private, stanno gradualmente popolando le orbite terrestri. Tuttavia, questo rappresenta sicuramente un problema a causa del sovraffollamento delle orbite principali. Queste orbite si stanno gradualmente riempiendo di oggetti inattivi, chiamati detriti spaziali, come satelliti non funzionanti e in avaria, stadi di veicoli di lancio abbandonati e detriti di frammentazione derivanti dalla rottura di corpi di razzi e veicoli spaziali in disuso.
Inoltre, è stato studiato da Donald J. Kessler che l'evoluzione incontrollata della popolazione di detriti spaziali porterà a una crescita a cascata causata dalle collisioni dei detriti. Ciò rappresenterà una grande minaccia per lo utilizzo delle orbite terrestri da parte delle future attività spaziali. Infatti, ogni collisione genererebbe un'enorme quantità di detriti, a causa dell'alta velocità dei satelliti e della conseguente alta energia dell'impatto, e ciò porterebbe a un effetto valanga che porterebbe a molte altre collisioni e a una crescita esponenziale dei detriti.
Per mitigare questo problema è necessario rimuovere un numero adeguato di detriti all'anno e contrastare così la crescita esponenziale del rischio di collisione. Una missione di rimozione attiva dei detriti ha l'obiettivo di individuare un gruppo di detriti e di farli deorbitare in un ordine definito.
Per selezionare una sequenza appropriata di detriti da deorbitare, si possono utilizzare diversi criteri di valutazione. La maggior parte della letteratura propone un'ottimizzazione bi-obiettivo per risolvere il problema, con l'obiettivo di minimizzare il ΔV totale e la durata totale della missione. Altri autori si concentrano anche sull'impatto dei detriti sull'ambiente orbitale. In questa tesi, la funzione da minimizzare tiene conto del costo totale, dell'impatto ambientale dei detriti e della difficoltà di cattura degli stessi, attraverso l'uso di due indici ponderati, l'indice ambientale (I_{ENV}) e l'indice di operabilità (I_{OP}).
Questo lavoro di tesi mira a sviluppare un algoritmo che trovi la migliore sequenza di detriti da deorbitare, considerando non solo il ΔV necessario, ma anche questi due indici.
In primo luogo, verrà introdotto il problema, parlando dei vari contributi della letteratura scientifica sull'argomento. Verranno poi analizzate le possibili architetture di missione, evidenziando vantaggi e svantaggi e parlando dell'architettura scelta per questa tesi. Verrà poi esposta la provenienza dei dati sui detriti utilizzati per la tesi, evidenziando la distribuzione dei detriti nelle diverse fasce orbitali e spiegando come è stata compensata la mancanza di alcuni dati.
A questo punto verrà esposto l'algoritmo vero e proprio. Dopo aver discusso il problema di ottimizzazione da risolvere e i diversi algoritmi che possono essere utilizzati, si spiegherà come è stato scelto un algoritmo Branch&Bound, che non considera solo il costo di trasferimento ma anche i due indici per determinare la migliore sequenza di deorbitaggio, e tutte le ipotesi e le assunzioni fatte. Alla fine, verranno mostrati e commentati i risultati ottenuti e verranno fatte alcune considerazioni sullo sviluppo futuro e sulle possibili implementazioni del codice.

Contents

List of Figures

List of Tables

Nomenclature

Acronyms

ADR	Active Debris Removal
AoP	Argument of Perigee
DKS	Deorbiting Kit Strategy
DOS	Disposal Orbit Strategy
ESA	European Space Agency
GEO	Geosynchronous Earth Orbits
GNC	Guidance Navigation and Control
$GTOC$	Global Trajectory Optimization Competition
$IADC$	Inter-Agency Space Debris Coordination Committee
LEO	Low Earth Orbit
PMD	Post-Mission Disposal
$RAAN$	Right Ascension of the Ascending Node
$TDTSP$	Time-Dependent Travelling Salesman Problem
TLE	Two-Line Elements
ToF	Time of Flight
TSP	Travelling Salesman Problem

Physics symbols

μ_E	Earth's gravitational constant
Ω	Right Ascension of the Ascending Node
ω	Argument of perigee
ω_a	Debris apparent angular rate
θ	True anomaly
a	Semi-major axis
E	Eccentric anomaly
e	Eccentricity

h	Orbit angular momentum
i	Inclination
J_2	Earth's first zonal harmonic
L_{MAX}	Debris maximum dimension
M	Mean anomaly
R_E	Earth's radius
S	Cross-sectional area
T	Orbit period

1. Introduction

1.1 Background

1.1.1 Debris problem

Nowadays, more and more companies are launching satellites into space, taking to an overpopulation of the Earth's orbits of interest, such as Low Earth Orbits (LEO) and Geosynchronous Earth Orbits (GEO). Since the first version of the Space Debris Mitigation Guidelines document in 2002, the Inter-Agency Space Debris Cordination Committee (IADC) settled some guidelines for the post-mission disposal in order to prevent the debris excessive growth. For a debris in LEO, it should be transferred into a disposal with an expected residual orbital lifetime of 25 years or shorter (although a direct re-entry is preferable). Moreover, the probability of success of the disposal should be at least 90%. Despite this, for many years satellites have been launched without taking this aspect into account. This has led to an endless number of non-operational satellites in the most populated orbits that could no longer be moved elsewhere, effectively becoming space debris as can be seen in Fig. 1.1.

Such a large number of uncontrollable satellites in the most populated orbits lead to a high risk of collision. This scenario was already predicted in 1978 by NASA consultant Donald J. Kessler. The astrophysicist hypothesized that, given the frequency of launches of new satellites into orbit, this would became so much saturated to take to a collision between satellites, which in turn would lead to an uncontrollable series of new collisions, making space completely inaccessible for any future mission [22]. In fact, it is worth noting that in LEO orbit objects can reach a velocity of the order of few km/s depending on the orbit. The energy generated by the impact would therefore be enormous even for objects of the size of a few centimeters, and a cloud of debris would then be released. This series of new debris could in turn be the cause of new collisions, leading to a sort of cascade effect known as the Kessler syndrome.

1.1.2 ADR missions necessity

In order to solve this problem and avoid all the catastrophic consequences of a possible collision, it will be necessary not only to implement all the end-of-life regulations of the satellite, but also to actively act to free the most used orbits from the objects with the greatest impact in terms of potential risk. Especially after the collision between Iridium 33 and Cosmos 2251 in 2009, there is a growing interest in using an ADR mission to mitigate the problem. Although an adr mission is demanding from various points of view (technical, economic, political etc.), it is clear that without it the environment of debris is destined to grow dangerously even in the absence of new launches. Liou [24] analyzed different scenarios considering both the case with no mitigation at all, the case with Post-Mission Disposal (PMD) only and the case with ADR with different removed objects per year. As can be seen in figure 1.2, without any mitigation plan debris is bound to grow exponentially, especially in LEO. Moreover, even with a PMD with 90% of success rate the situation will be unsustainable in the future. Only by combining it

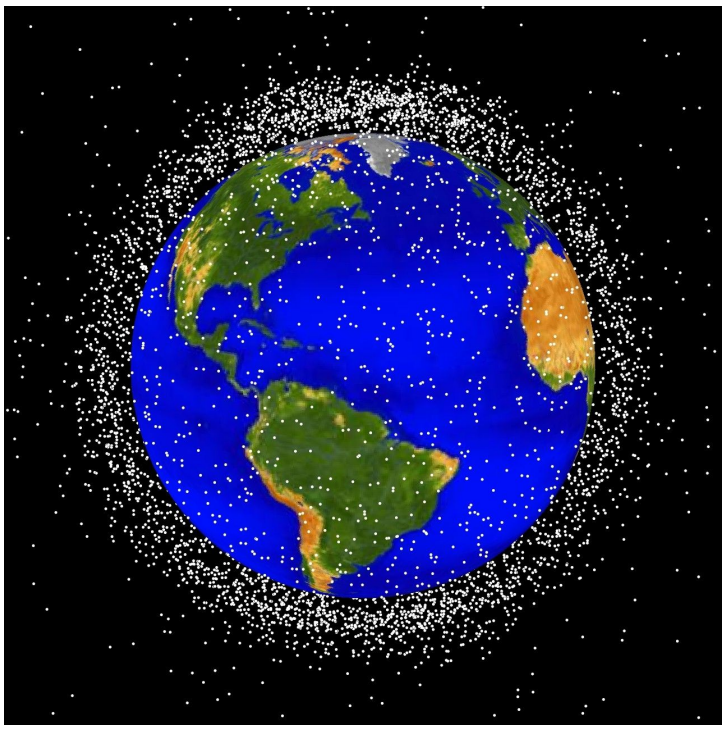
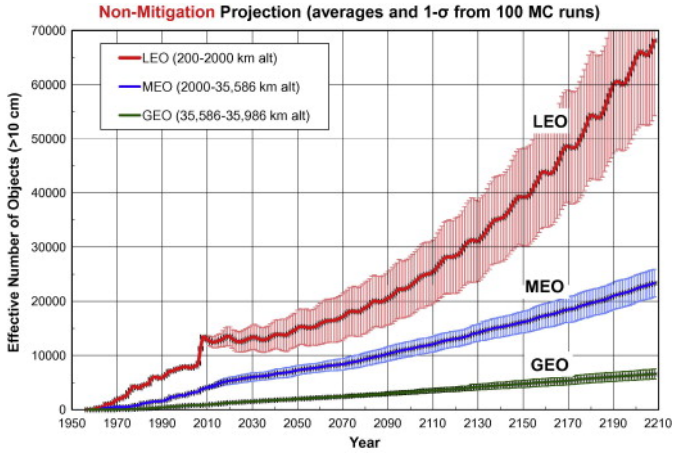
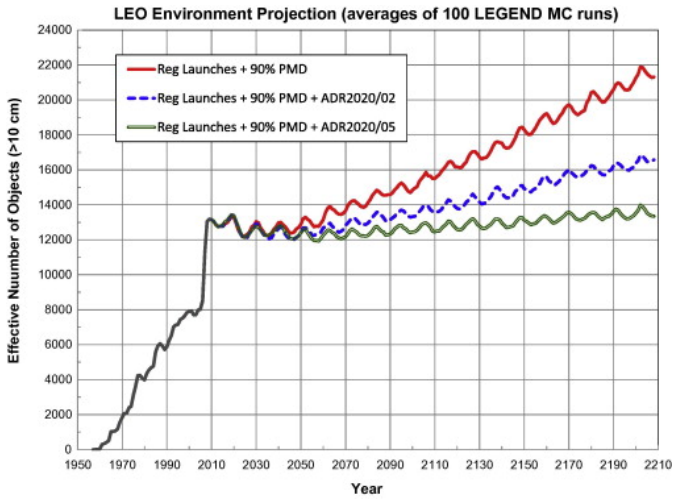


Figure 1.1: Debris located in LEO orbits (within 2,000 km of the Earth's surface) plotted by NASA in 2005 [40]. The orbital debris dots are scaled according to the image size of the graphic to optimize their visibility and are not scaled to Earth.

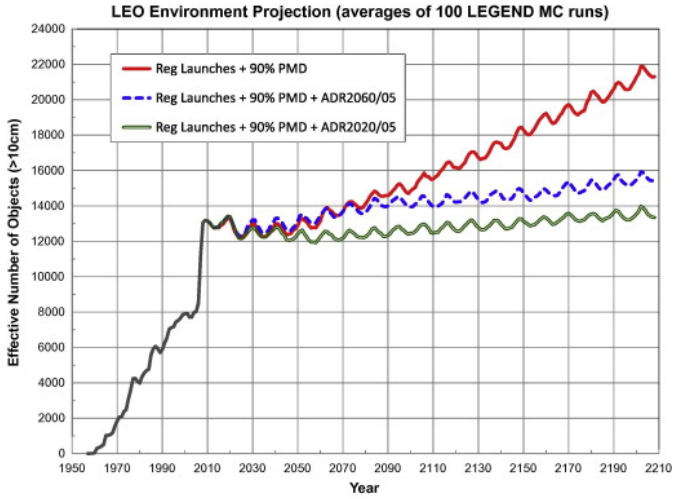
with an Active Debris Removal (ADR) mission with 5 deorbited debris per year a satisfactory result can be obtained. As can be seen in the last plot, the sooner we start to clean Earth's orbits with appropriates ADR missions, the better it will for the future of the space missions.



(a) Non-mitigation projection for LEO, MEO and GEO



(b) LEO environment projection with a 90% success rate PMD alone and PMD combined with ADR starting from 2020 with 2 and 5 debris deorbited per year



(c) LEO environment projection with a 90% success rate PMD alone and PMD combined with ADR starting from 2020 and 2060 with 5 debris deorbited per year

Figure 1.2: Future growth simulation of orbital objects (> 10 cm) for different test scenarios computed by J.-C. Liou [24].

1.2 State of the art

1.2.1 Actual ADR missions

Given the enormous technical difficulties, especially in debris docking operations, a satellite with this objective has not yet departed, although studies on this are increasingly numerous. The European Space Agency (ESA) in 2013 started to work on e.Deorbit, a mission with the task of capturing and safely deorbiting a derelict ESA-owned satellite, the Envisat Earth-observing satellite, in highly trafficked LEO [2]. The deorbiter would have had to attach itself to the debris through a robotic arm, and then re-enter with the debris itself. However, the funding of the mission stopped in 2018. Instead, ESA member states funded the ClearSpace-1 mission, which is now under development. The aim of this mission is to remove a 112 kg debris, a Vega upper stage, in a similar way to the e.Deorbit mission, however using 4 robotic arms instead of one. The launch is planned for 2025, and this would be the first debris removal mission ever [1].

1.2.2 Removal techniques

There are several ways in which a debris can be deorbited, and they can be divided into three types: contactless, rigid contact and non-rigid contact. First ones could be ion beam or chemical shepherd that lower the perigee of the debris orbit. In second one, the chaser have to phisically dock to the debris. After that, he can decide to take the debris with him into a disposal orbit, or to attack to it some devices (like de-boost engine kit, tethers or drag augmentation devices) that make it lower up to a disposal orbit. The last typology include mechanisms like harpoon or nets to capture the debris. However, the dynamics would be very complex. Colmenarejo et al. [13] summed up all the different devices and the Guidance Navigation and Control (GNC) aspect related to them.

1.2.3 Sequence optimisation for multiple target mission

The other aspect to consider in an ADR mission that was largely studied during last decades is the optimization of the sequence of debris to deorbit, treated as a Time-Dependant Travelling Salesman Problem (TDTSP). The 9th edition of the Global Trajectory Optimization Competition (GTOC) was dedicated to this problem too [20]. Bérend et al. [4] proposed a bi-objective optimization using a Branch&Bound method, considering as objective function to minimize the total propellant consumption, with the simplification of having all Hohmann transfers, and the total mission duration. That algorithm allows to find a good estimation of the Pareto front with an acceptable computational cost for a 10 debris removal scenario. Madakat et al.[28], similarly, used a Branch&Bound methode for a bi-objective simplification to deorbit 5 debris per year. However, he computed the manoeuvres solving the Lambert's problem. Casalino et al. [10] used an evolutionary algorithm to determine the actual legs of the most promising opportunities of multiple-target mission. The rendez-vous time and the cost is estimated through a fast procedure exploiting the effect of Earth's oblateness to find the time windows in which the RAAN matched. Cerf [11] defined a specific transfer energy with impulsive manoeuvres so that the problem becomes of finite dimension, in order to successively linearizing it around an initial reference solution. Then he used a Branch&Bound algorithm to optimize simultaneously the debris sequence and the orital manoeuvres. He found that the optimal solution is very close to the initial guess. Di Carlo et al. [9] used a bio-inspired evolutionary algorithm, known as Physarum algorithm, to optimize the selection of 10 uncooperative debris to deorbit in 1 year. Moreover, like Huang [18] and Braun et al. [6], she used a low-thrust model for the transfers. In fact, using electric propulsion instead of chemical would result in a lower propellant usage. However, the dynamics and the computation beyond electric propulsion would be far more complicated. Hokamoto et al. [31] and Liu et al. [26] proposed a genetic algorithm (ga), an evolutionary algorithm largely used for NP-hard problems like this one, to solve the optimization problem,

using an adequate fitting function related to the transfer cost. Shen et al. [38] proposed to solve the problem using an Ant Colony Optimization (ACO), an evolutionary algorithm that, similarly to genetic algorithm and Physarum algorithm, make use of a fitting function to guess the quality of the solution. Masserini in his thesis [29] tested several algorithms, namely the brute force algorithm, nearest neigbour algorithm and branch&cut algorithm, for an ADR mission to remove failed satellites from a large satellite constellation, comparing their performance with the solution obtained from a Monte Carlo simulation.

1.2.4 Problem of candidates ranking and selection

It worth noting that all the solutions discussed above aim to optimize the problem of debris and sequence selection mainly in function of the ΔV (or propellant consumption) and the time. However, deorbiting debris without considering their characteristics and their impact on the environment could not be the best choice. For this reason, there are studies that take this aspect too into account. For example, Peterson [35] introduced a parameter defined as "probability-severity" (P-S). It takes into account the probability of collision, related to the size of the debris and to the spatial density of the other object in the same orbital region, and the severity of the collision, related to the kinectic energy (mass and velocity) of the debris. Objects with an higher P-S value will be more likely to be chosen. In a similar way, Borelli et al. [5] proposed a function to minimized that is the weighted sum of four different indices: Environmental Index, Economical Index, Operability index and Mission Related Index. The first one describe the criticality of a certain inactive object to the debris orbital environment, the second one quantified the economic resource value endangered by each debris object, the third one describes the operative requirements and complications arising during the operations of approach and capture of a uncooperative target and the last one is related to the total ΔV of the ADR mission. The Environmental and the Operability indices are the ones I use in my thesis dissertation.

1.3 Scope of the thesis

The aim of this thesis is to implement a mission designer for a multiple-target ADR mission that will find the best route in order to remove the required number of debris. The algorithm will take account not only of the ΔV necessary for each travel, but also of many characteristics of the debris that influence the capture feasibility and its impact on the orbital environment. To do that, they will be used some specific indices, the Operability Index and the Environmental Index, that will be discussed in their dedicated paragraph in Section 4.4. and 4.5. The transfer cost and these two indices will be weighted together in order to find a solution that will be a compromise between these three factors. Anyway, the weight of the indexes can be tuned by the user of the code according to the needs and mission objectives.

1.4 Thesis organisation

After the introduction, my thesis is divided into five chapters.
In Chap. 2 it will be explained the mission architectures, like the OTV characteristics, the deorbiting and the transfer strategy, the type of propulsion and the removal techniques, considered in my dissertation, with a brief digression on all the possible strategies and their impact on the code.
In Chap. 3 the available data on debris will be analyzed. Here, the provenance of the data, the location of debris in Earth's orbits and the solution to the problem of missing data will be discussed.
In Chap. 4 it will be described the algorithm for the debris sequence optimization. First, the

problem will be introduced and its possible solution will be briefly discussed. Then, the focus will shift to the solution adopted in this dissertation. The solving algorithm and the related indices used for the function to minimize will be discussed.
In Chap. 5 the simulation scenarios and their results will be presented and commented.
Finally, in Chap. 6 the results of the simulation will be discussed, and future works to improve the quality of the solution will be proposed.

2. Mission architectures

2.1 Architectures overview

In order to fulfill the goal of this dissertation, different strategies can be adopted. Each equally effective, and each with its pros and cons. The overall possible architectures will be introduced hereafter, with a focus on their implication on the mission planning algorithm for the ADR. A summary flowchart is showed in Figure 2.1.

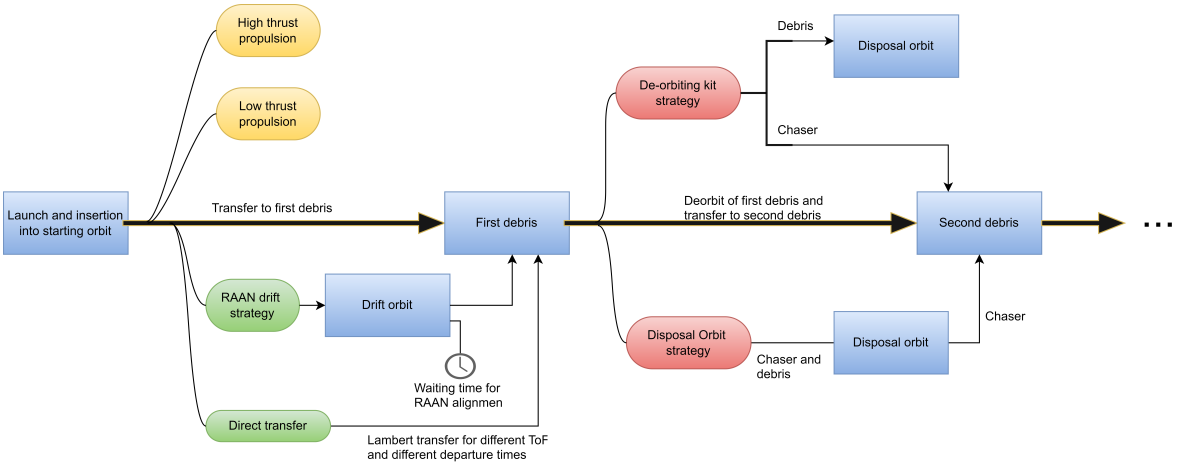


Figure 2.1: ADR mission deorbiting phases flowchart considering different architectures

2.1.1 RAAN change strategy

Among the various orbital transfer maneuvers, those out-of-plane are certainly the most expensive. As for the change of inclination, the cheapest possibility is to provide an impulse perpendicular to the plane at the apogee. As for the Right Ascension of the Ascending Node (RAAN) change, there are two ways to compensate for the difference between the departure orbit and the arrival orbit.
The simplest way is the direct change through an out-of-plane impulse. This manoeuvre, unlike the other strategy, results in a transfer with small ToF. However, it is very expensive and it could preclude lots of transfers with a large RAAN difference.
An alternative strategy takes advantage of the RAAN variation due to the Earth's oblateness effects, particularly to the second zonal harmonic J_2. In fact, the secular rate of change of RAAN due to J_2 effects can be expressed as in Equation 2.1.

$$\frac{d\Omega}{dt} = -\frac{3}{2} \cdot J_2 \cdot \left(\frac{R_E}{a(1-e^2)} \right)^2 \cdot \sqrt{\frac{\mu_E}{a^3}} \cdot \cos i \tag{2.1}$$

The orbital parameters that influence the RAAN change are the semi-major axis and the inclination. To reduce a large RAAN difference between initial and final orbit, the transfer can

be performed using an intermediate *drift orbit*, with specific semi-major axis and inclination. Specifically, the time needed to compensate a RAAN difference can be expressed as in Equation 2.2.

$$\Delta t_{DRIFT} = -\frac{\Omega^0_{DRIFT} - \Omega^0_{DEBRIS}}{\dot{\Omega}_{DRIFT} - \dot{\Omega}_{DEBRIS}} \tag{2.2}$$

where Ω^0_{DRIFT} and Ω^0_{DEBRIS} are the initial values of RAAN for the chaser in the drift orbit and of the debris, while $\dot{\Omega}_{DRIFT}$ and $\dot{\Omega}_{DEBRIS}$ are the RAAN variation of the drift and the debris orbit. Depending on the inclination of the orbits and the difference in RAAN, the drift orbit will have an higher or a lower semi-major axis. In fact, for orbit with inclination up to 90°, the change in Ω will be always negative. Moreover, the greater the semi-major axis is, the smaller the variation will be in absolute terms.

2.1.2 De-orbit strategy

In order to be safely deorbited, the debris has to be taken into a de-orbiting orbit. More specifically, the perigee of the debris orbit must be lowered to an altitude that results in a reentry time lower than 25 years [43]. In fact, due to the atmospheric drag, the apogee will continue to decrease, as can be seen in Figure 2.2, until the debris returns to earth within a set maximum return time. There are mainly two strategies by which the chaser can deorbit the debris.

The first one is the Deorbiting Kit Strategy (DKS). This strategy consists in attaching to the

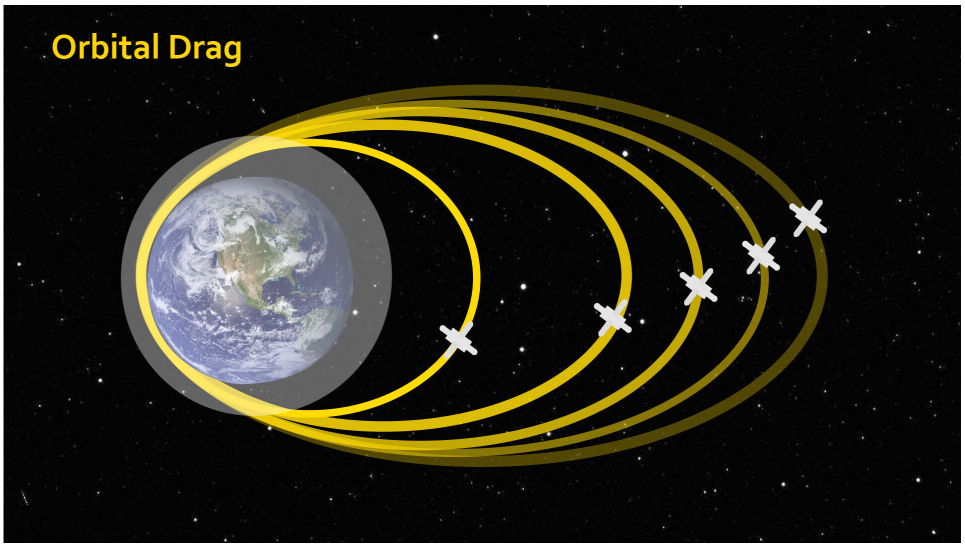


Figure 2.2: Orbit perturbation due to atmospheric drag (image from [32])

debris a deorbiting kit, like a thruster, that will take the debris into the disposal orbit. The other one is the Disposal Orbit Strategy (DOS). In this case, the chaser attaches itself directly to the debris and takes to the disposal orbit. For both strategies, there are different removal techniques that can be adopted, like described in 1.2.2.

DKS is more efficient in terms of deorbiting phase duration, since the chaser can depart for the new debris right after attaching the kit, while in DOS it is necessary to wait for the chaser to change the orbit of the debris. Moreover, the propellant used to deorbiting the kit is lower in DKS since the kit have to deorbit the debris only, while in DOS it has to deorbit the chaser too. However, having to attach a detachable kit can lead to more complicated operations to capture and secure the kit attachment to the debris.

2.1.3 Propulsion type

The different choice of thrusters and propellant will influence the dynamics of orbital transfers, the propellant consumption, the transfer duration and computational cost. It is possible to mainly distinguish between two types of propulsion. A summary scheme can be seen in Figure 2.3.

High-thrust propulsion is the one that we have in chemical propulsion systems. It uses

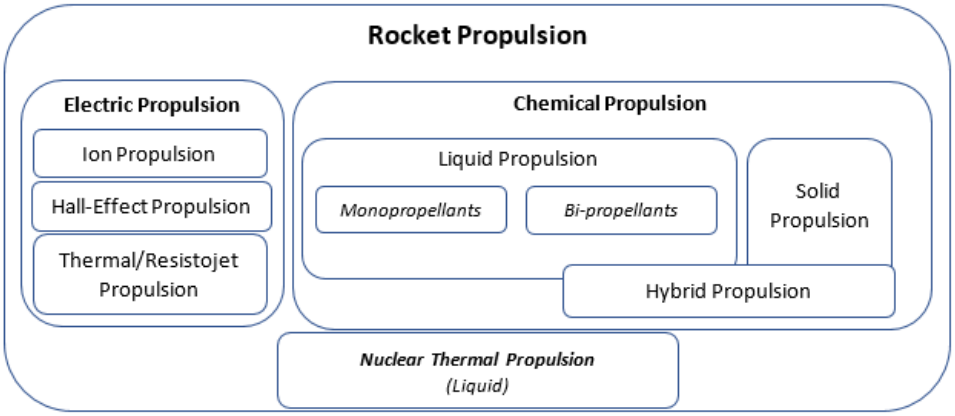


Figure 2.3: Breakdown of different types of rocket propulsion (image from [12])

chemical reactions to release energy and accelerate gases to generate thrust. These systems produce relatively high thrusts in short periods of time. There are several kinds of chemical propulsion, including liquid/gaseous propulsion, solid propulsion, and hybrid propulsion. The manoeuvres carried out by a chemical thruster can be considered to be impulsive. Impulsive manoeuvres are such that can be modelled as instantaneous change of magnitude and direction of the orbital velocity vector. The propellant consumption for impulsive manoeuvres is described by the Tsiolkovsky Rocket Equation in Equation 2.3.

$$\frac{\Delta m}{m} = 1 - e^{\frac{\Delta v}{I_{SP} \cdot g_0}} \tag{2.3}$$

where Δm is the propellant mass used for the manoeuvres, m is the initial mass (satellite and propellant), Δv is the transfer cost, I_{SP} is the specific impulse of the thruster and g_0 is the standard acceleration due to gravity at sea level.

Low-thrust propulsion is, instead, the one that we have in electric propulsion system. It typically uses electric heating or electric or magnetic fields to accelerate propellants (usually gases). Thanks to the high exhaust velocities, and therefore to the high specific impulse, these systems can be very fuel-efficient but can only accelerate relatively few particles of gasses at a time, resulting in very small thrusts. In this case, the manoeuvres will no longer be impulsive, and that means that the spacecraft will change position during the manoeuvres, and it will typically perform several revolutions before reaching the desired orbit. The resulting trajectory will be a sort of spiral, like the one showed in Figure 2.4. Moreover, during the manoeuvres it will be also necessary to take account of the variation of orbital parameter due to the low-thrust perturbing acceleration and of the secular effects due to J_2.

2.1.4 Orbital manoeuvres

Knowing the initial position and the target position, there are two ways that can be considered to simplify impulsive transfers and thus compute the transfer cost. The first way is mostly a simplification that permits to reduce the computational cost and to ignore the dependence from the Time of Flight (ToF). It consists in performing an Hohmann transfer, an orbital inclination change and a phasing manoeuvre. Hohmann transfer is a low-energy transfer between two

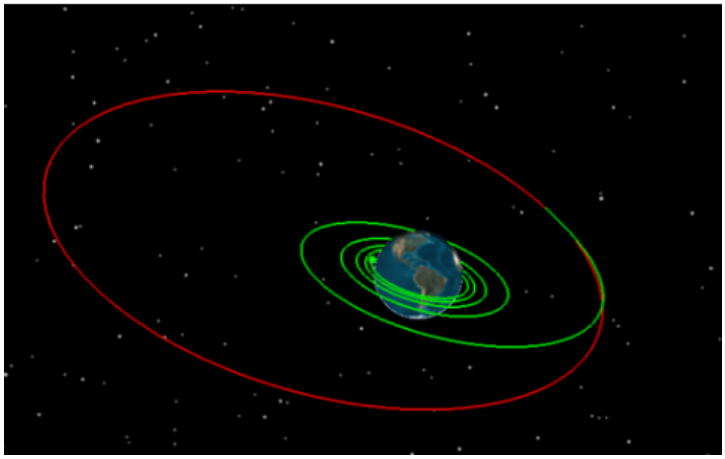


Figure 2.4: Example of low-thrust spiral trajectory (image from [27])

coplanar orbits, and it is accomplished by placing the spacecraft into an elliptical transfer orbit that is tangential to both the initial and target orbits, as can be seen in Figure 2.5. The

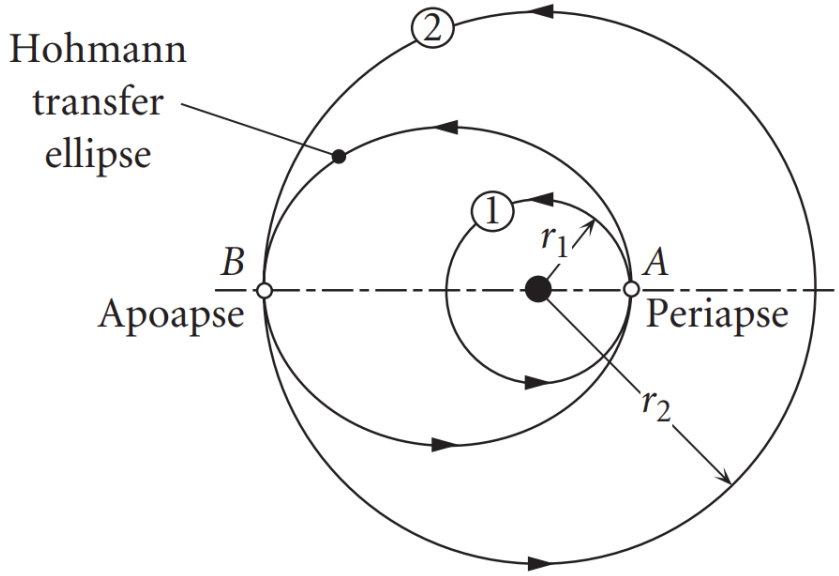


Figure 2.5: Hohmann transfer between coplanar orbits (image from Curtis [14])

maneuver uses two impulsive engine burns: the first establishes the transfer orbit, and the second adjusts the orbit to match the target. In case of elliptic orbits, it will be necessary to choose an adequate point of departure and arrival to obtain the most efficient transfer. In general, for a transfer from a lower to an higher altitude, it is most efficient for the transfer orbit to begin at the periapse on the inner orbit, where its kinetic energy is greatest, regardless of shape of the outer target orbit. If the inner orbit is circular, the transfer ellipse should terminate at apoapse of the outer target ellipse, where the speed is slowest. In the opposite case, from an outer circle or ellipse to an inner ellipse, the most energy-efficient transfer ellipse terminates at periapse of the inner target orbit. If the inner orbit is a circle, the transfer ellipse should start at apoapse of the outer ellipse. In order to compute the total cost for the Hohmann transfer, first it is necessary to compute the angular momentum of all the three orbits using Equation 2.4:

$$h = \sqrt{2\mu_E}\sqrt{\frac{r_a r_p}{r_a + r_p}} \tag{2.4}$$

where r_a and r_p are the apogee and the perigee radius respectively. Then, it is possible to calculate the velocity at each point using Equation 2.5:

$$v = \frac{h}{r} \tag{2.5}$$

At this point, the total cost is given by the difference in absolute terms between the orbits velocities and the velocity of the Hohmann transfer orbit, as shown in Equation 2.6:

$$\Delta v_{HOH} = |v_{HOH,A} - v_{1,A}| + |v_{HOH,B} - v_{2,B}| \tag{2.6}$$

where, referring to fig. 2.5, $v_{HOH,A}$ and $v_{HOH,B}$ are the velocities of the Hohmann transfer orbit in point A and B respectively, $v_{1,A}$ is the velocity of orbit 1 in point A and $v_{2,B}$ is the velocity of orbit 2 in point B.
The orbital inclination change is an out-of-plane manoeuvre, that means that the velocity change vector is perpendicular to the orbital plane. It is always convenient to perform this impulse when the tangential velocity is lower, that is at the apogee of the orbit with the higher apogee radius. The transfer cost is computed using Equation 2.7:

$$\Delta v_{INC} = \left| 2v \sin \frac{\Delta i}{2} \right| \tag{2.7}$$

where v is the tangential velocity and Δi is the difference in the inclinations of the two orbits. The phasing manoeuvre is necessary to allow the rendez-vous between two objects placed in the same orbit. It is, in general, a low cost manoeuvre and can be neglected for a preliminary analysis. Firstly, it is necessary to compute the difference in time between the two objects knowing their true anomaly θ. To do that, the time of flight from $\theta = 0°$ need to be computed using Equations 2.8 and 2.9:

$$E = 2 \arctan \left(\sqrt{\frac{1-e}{1+e}} \tan \frac{\theta}{2} \right) \tag{2.8}$$

$$t = \frac{T}{2\pi}(E - e \sin E) \tag{2.9}$$

Then it will possible to know the difference in time between two objects, and then computing the period of the phasing orbit subtracting or adding (depending on the relative position of the two objects) this time to the starting orbit period, from which the semi-major axis of the phasing orbit can be computed as:

$$a = \left(\frac{T\sqrt{\mu}}{2\pi} \right)^{\frac{2}{3}} \tag{2.10}$$

However, although this is still a good simplification for a preliminary analysis, the most correct and realistic way to calculate a maneuver between two points, knowing their position vectors, would be by solving the Lambert's problem. The problem is geometrically represented in Figure 2.6. According to a theorem of Lambert [14], the transfer time Δt from P_1 to P_2 is independent of the orbit's eccentricity and depends only on the sum $r_1 + r_2$ of the magnitudes of the position vectors, the semi-major axis and the length c of the chord joining P_1 and P_2. Since $\mathbf{r}_2$ and $\mathbf{v}_2$ can be expressed as:

$$\mathbf{r}_2 = f\mathbf{r}_1 + g\mathbf{v}_1 \tag{2.11}$$

$$\mathbf{v}_2 = \dot{f}\mathbf{r}_1 + \dot{g}\mathbf{v}_1 \tag{2.12}$$

which yelds to

$$\mathbf{v}_1 = \frac{1}{g}(\mathbf{r}_2 - f\mathbf{r}_1) \tag{2.13}$$

$$\mathbf{v}_2 = \frac{1}{g}(\dot{g}\mathbf{r}_2 - \mathbf{r}_1) \tag{2.14}$$

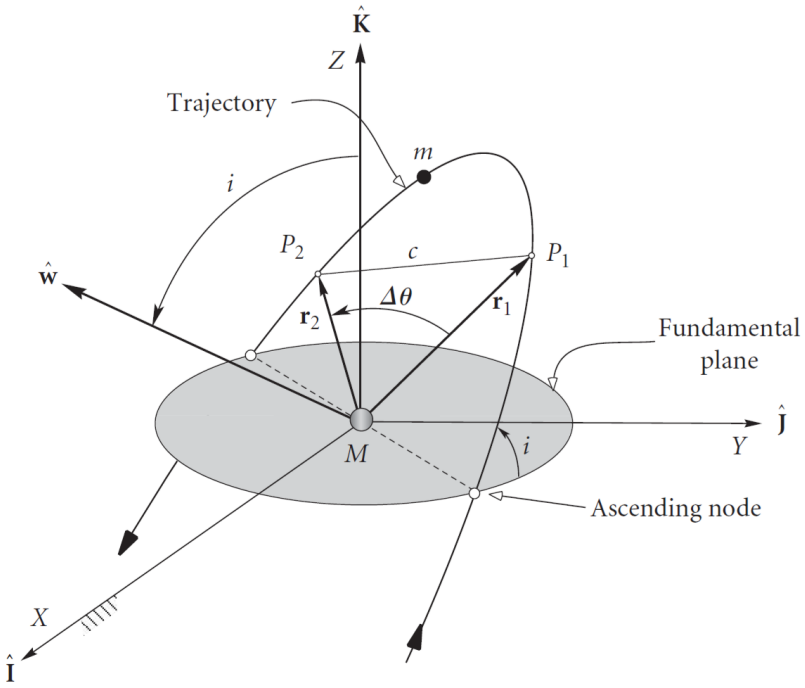


Figure 2.6: Lambert's problem (image from Curtis)

where g, f, $\dot{g}$ and $\dot{f}$ are the Lagrange coefficients that will be determined with an appropriate method [14].

2.2 Selected mission architecture

In this dissertation, the mission architecture for active debris removal of multiple target selected is the DKS. In fact, as explained above, it is more efficient in terms of mission duration and propellant usage. The chaser will carry a number of kit equal to the number of the debris to deorbit. In the scenarios considered in this study, the de-orbiting kit is a 100 kg-solid rocket stage.
For the sake of simplicity, it was chosen to adopt the direct change of RAAN. This strategy may take to less optimal solutions with respect to the drift orbit strategy, but it is computationally less expensive and it permits to save a lot of time. However, for each travel it is considered a mesh of different departure times and ToF, in order to allow the code to find more favorable combinations.
For all the transfers between debris the transfer path and the costs are computed using an algorithm that solves the Lambert problem using the Battin formulation [3]. Regarding the de-orbiting manoeuvres, instead, they are considered to use a semi Hohmann transfer that will simply lower the perigee of the debris orbit up to 400 km.
The adopted propulsion system is an high-thrust propulsion, and then all the transfers are considered to be impulsive. This choice permits to reduce the computational time, since low-thrust transfers lead to a very complex dynamics, and then is acceptable for a preliminary analysis. Low-thrust transfers can be considered in later phases of the mission, to optimize one of the sequences already identified by the algorithm in the preliminary phase.

3. Debris analysis

3.1 Data source

All the debris data are taken from CelesTrak [21] and DISCOSweb [15] databases. More specifically, from Celestrak they are taken the data regarding the orbit, given in Two-Line Elements (TLE) format. Then, using their identification number, DISCOSweb was used to extract data regarding their mass and geometrical properties. Moreover, the data regarding the angular rate are obtained from light curves data, as explained in 3.3.1, which were taken from MMT Observatory database [19]. From all the available data, the ones related to inactive payloads and to rocket bodies are considered. All these data were then all gathered in a Matlab structure containing all the debris and the respective information, such as the id, the cross-sectional area, the mass, the object class, the dimensions, the shape, the angular rate, the rotational state, the percentage in eclipse and the information on the orbit, i.e. the semi-major axis, the eccentricity, the inclination,the argument of perigee, the right ascension of the ascending node and the eccentric anomaly. Data from CelesTrakare are given in Two-Line Elements (TLE) format. The debris orbital parameters are extracted from the file containing all TLE and saved in the Matlab structure in order to make them more handy and accessible. All the available data were updated the 10th of April of 2022.
Since the selected databases contain data of 7917 debris, it was necessary to restrict the search field to a portion of them. Since the algorithm will prefer transfers with a low cost, the debris will generally located inside a limited bin of inclination and semi-major axis. In order to choose appropriate bins, it will be useful to look at the debris location as did in 3.2. Moreover, in order to narrow the search, a minimum value of the mass of the debris to be deorbited can be defined. In fact, it is more useful to remove the heavier debris since their eventual collision would generate a much higher quantity of debris and also since they would have a greater probability of collision given the larger size.

3.2 Debris location

The distribution of all the debris from available data is plotted in fig. 3.1. Each bin covers a change of 800 km in altitude and 5° in inclination. The color represent the number of debris located in each bin, as showed in the colorbar on the side. Dark blue bins are generally populated by one or at least a few debris. As can be seen, the majority of space debris is located in LEO. These orbital region are by far the most exploited for commercial and scientific missions. Moreover, even for higher altitude missions, the rocket bodies used to inject spacecraft on the operational orbits will reside in the LEO region after injection. It is possible to see a more detailed view of the LEO region in Figure 3.2, with a bin range of 50 km in altitude and 4° in inclination, and at its side the debris flux is displayed. This parameter was obtained using ESA's MASTER-8 space debris simulation environment [16] for debris with a diameter greater than 10 cm. It is possible to identify some areas more populated by debris where there is more

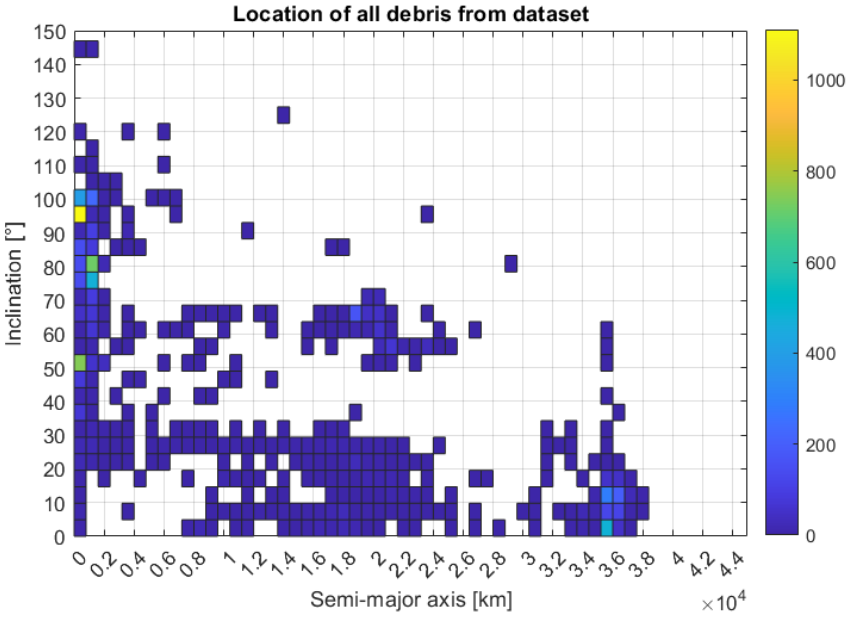
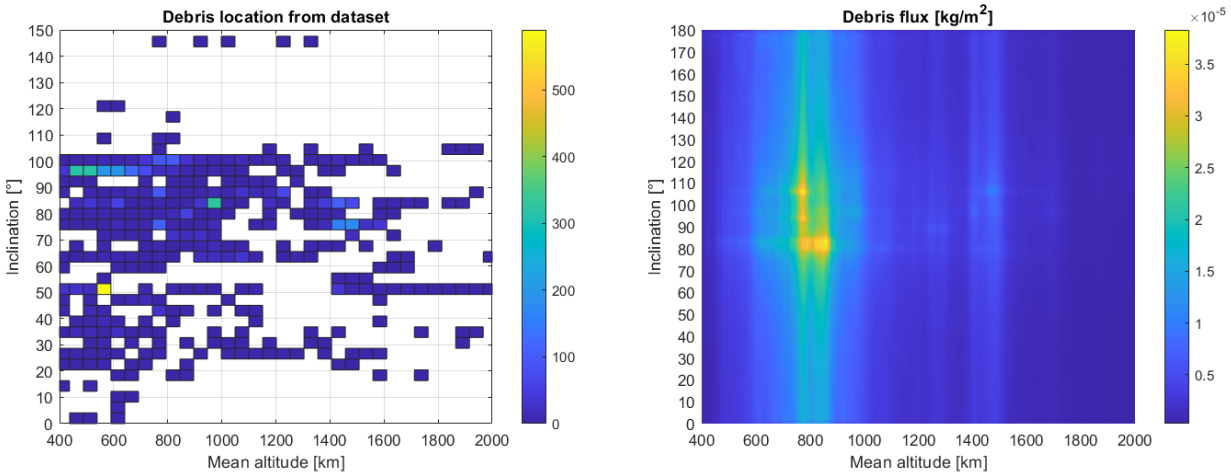


Figure 3.1: Total debris distribution based on semi-major axis and inclination



(a) Debris location in LEO

(b) Debris flux in LEO

Figure 3.2: Debris distribution and debris flux in LEO

need to intervene:

- The LEO sun-synchronous area with an altitude up to 2000 km and and inclination near 90°
- The LEO area with an inclination near 50°, mainly populated by telecommunication satellites in constellation
- The GEO area, with an altitude of about 36000 km and with no inclination

Since taking account of all possible debris in the preliminary analysis would be computationally demanding, it would be reasonable to consider some areas with limited bins of semi-major axis and inclination. In fact, transfers with a large difference in a and i would be very expensive and therefore will not be taken into consideration.

3.3 Missing data

Since for a large amount of debris there are some unavailable data, it was necessary to assume some of them and eventually neglected these debris in case of missing of fundamentals data.
In case of missing data regarding the debris orbit it is impossible to do any assumption. Then, all these debris are neglected. The only missing data allowed is the argument of perigee ω. In fact, since all the orbit are almost circular, the compensation for the difference in ω is almost negligible. The default value is then set to $\omega = 0°$.
For debris with missing mass information, the conventional value of 100 kg was chosen. In fact, it is reasonable to think that the debris for which such information is not known are small debris derived, for example, from fragmentation. With regards to the cross-sectional area S, in the absence of information regarding the size of the debris, it was decided to assign a value according to the mass of the debris. This value was obtained from the normal distribution of the cross-sectional area of the debris on which we had information, for different mass classes, as can be seen in Figure 3.3. The other assumption made for the missing data concerns the

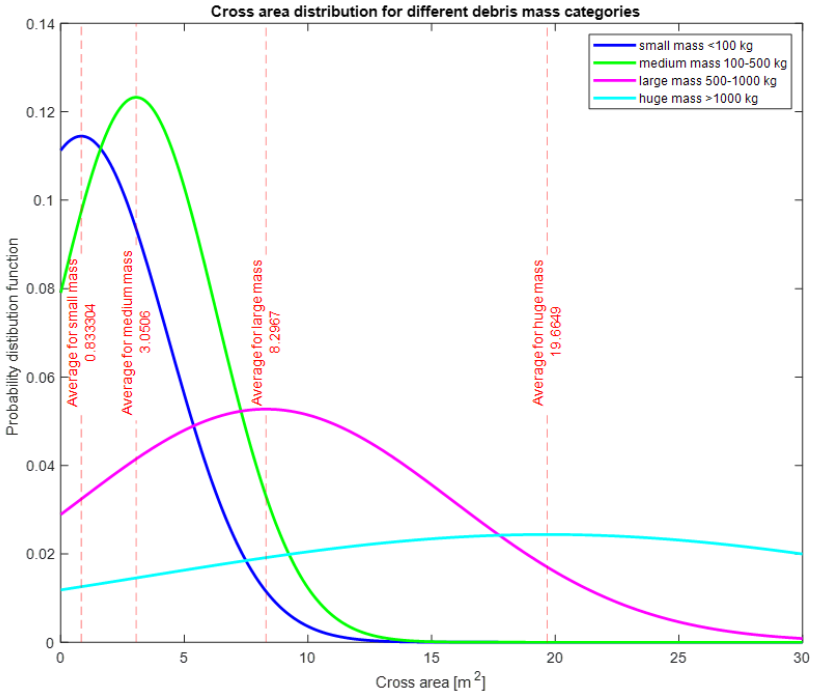


Figure 3.3: Cross-sectional area values distributions for different mass classes

percentage of the orbit in eclipse. This parameter is conventionally set to 0.7263 for debris for which it is not specified, according to the distribution showed in Figure .

3.3.1 Apparent angular rate

The last data for which a standard value has been established is that of the apparent angular rate ω_a. The apparent angular rate is the speed at which the debris appears to spin. In fact, not being able to know their real angular velocity, what one does is try to establish it by looking at their light curves. Light curves are graphs of the light intensity of an object over the time. Looking at them, it is possible to identify the frequency with which a shimmer is detected when looking at the debris [39]. This frequency should be related to how fast the debris is spinning. In Figure 3.5 it can be seen how this light is acquired by a telescope while in Figure 3.6 the

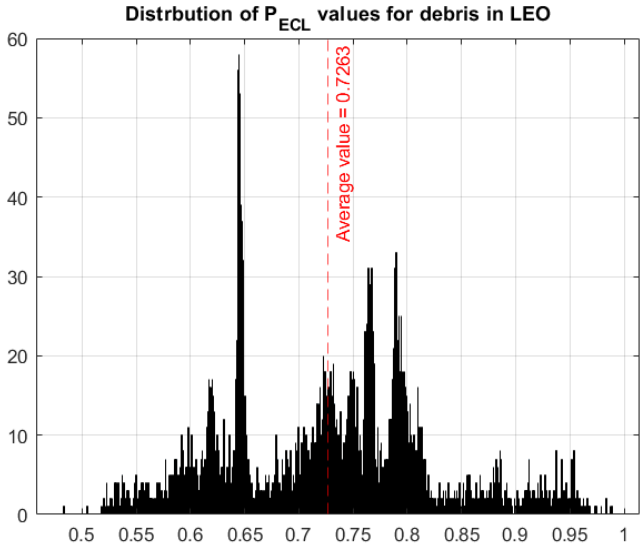


Figure 3.4: Distribution of P_{ECL} values for debris in LEO

resulting graph of the debris reflected light magnitude over time, from which the apparent angular rate will be extracted.

The attitude state of a debris can be considered to be periodic or aperiodic, whether or not

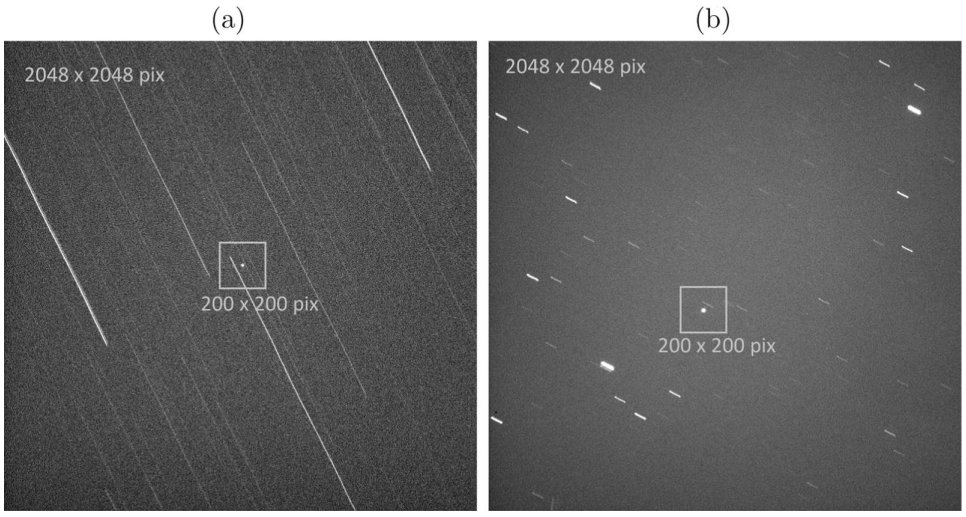


Figure 3.5: Full frames acquired by ZIMLAT telescope for satellites Swisscube (2009-051B) (a) and GLONASS COSMOS 2380 (1995-009B) (b) during the object tracking mode. Used exposure times were 0.5s and 1.0s, respectively. (from [39])

it shows a pattern in the light curve. In case of missing data regarding the apparent rotational state, the default value for debris with periodic attitude state is $\omega_a = 1°/s$, while for aperiodic debris the default value is $\omega_a = 3°/s$

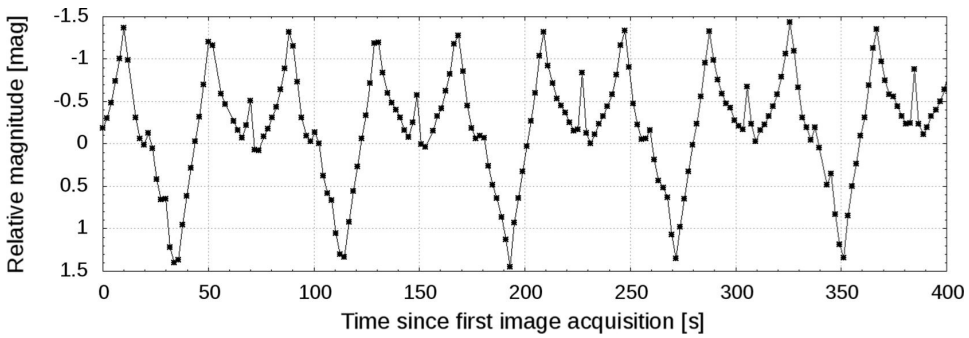


Figure 3.6: Example of an acquired light curve for object 2001-053C with clear periodic signal. (from [39])

4. Debris sequence optimization

4.1 Time-Dependent Travelling Salesman Problem

One of the main challenges related to ADR missions is finding the best path to deorbit a certain number of debris, optimizing a certain cost function. This can be reduced to a discrete sequence/path optimisation problem similar to the Traveling Salesman Problem (TSP), widely known in literature.
As suggested by the name, TSP is based on the problem regarding the best path that a salesman should do to visit a certain number of cities using as little fuel as possible, starting from his current city. A generic scheme for the TSP is showed in Figure 4.1. Each node represent a city that have to be visited, while the lines represent the possible paths between cities, where the number is the cost (kilometers or fuel consumption) of the travel between two cities. The scope of the TSP optimization is to visit all the nodes while minimizing this cost.
Logically, the classic TSP shows some huge differences with the ADR sequence optimization

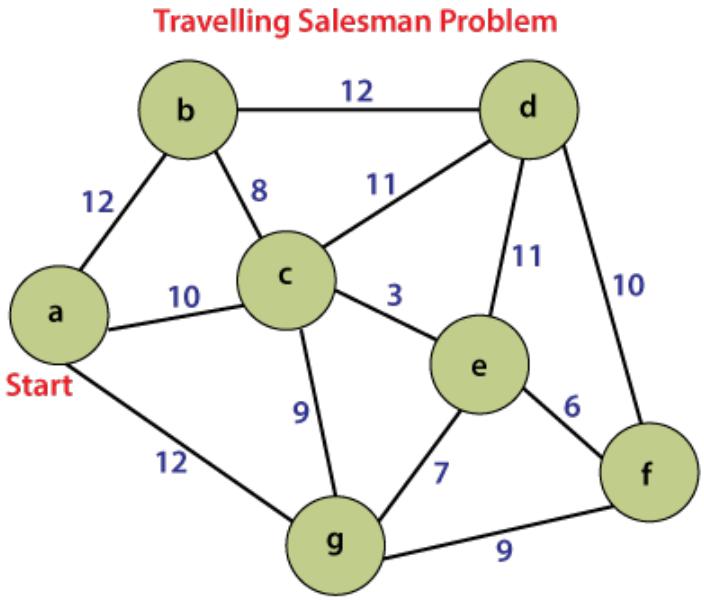


Figure 4.1: Example of a generic TSP scheme. (from [41])

problem. First of all, there are no cities, but in their place there are space debris that have to be reached by the OTV. Unlike cities, space debris change their position with time, and then the transfer cost between them will depend on their actual position and then on the time. This more general case of TSP is known as Time-Dependent Travelling Salesman Problem (TDTSP), and it is in general more complex since it is necessary to take into account the displacement of the nodes over time.

4.2 Optimization algorithms for NP-hard problems

The TSP falls under combinatorial optimization, which deals with problems in which all or part of the unknowns are integer variables. The classification of combinatorial problems lies on the existence or not of polynomial-time algorithms, on one hand to solve the problem, on the other hand to check a solution [23]. An algorithm is said to be of polynomial time if its running time is upper bounded by a polynomial expression in the size of the input for the algorithm, that is $T(n) = O(n^k)$ for some positive constant k [33]. P-class problems have a known polynomial time solving algorithm that can be used to check the solution too. NP-class includes problems which have a known polynomial time checking algorithm, but an unknown solving algorithm. As NP plays a central role in computational complexity, it is used as the basis of several classes. The TDTSP is an NP-hard problem. NP-hard is a class of problems which are at least as hard as the hardest problems in NP. For these combinatorial problems, the resolution methods fall into three main categories [11]:

- Explicit enumeration
- Implicit enumeration algorithms
- Evolutionary algorithms

4.2.1 Explicit enumeration

Explicit enumeration simply consists on a brute force algorithm that evaluates all the possible combinations and find the one with the lowest cost function. Obviously this method would be impracticable for a large problem. In fact, the total number of arrangements of n debris amongst N is $A_N^n = \frac{N!}{(N-n)!}$. Considering a population of 76 debris and 4 debris to be deorbited, like in this dissertation, would lead to compute the cost function for more than 30 millions combinations. Moreover, considering d different departures times from each debris and f different ToF for Lambert transfer, the number of arrangements would increase drastically and would become $A_N^n = \frac{N!\, d^{n-1}\, f^{n-1}}{(N-n)!}$. Then, using 6 different departures times and 36 different ToF would take to more than 10^{16} combinations.
Despite that, explicit enumeration can be still used to validate a more complex algorithm considering a much smaller population than desired, making it possible to compute the cost function for all possible combinations. The optimal result can then be compared with the one found using the algorithm that need to be tested.

4.2.2 Implicit enumeration algorithms

Implicit algorithms like Branch&Bound and Branch&Cut explores all the possible combinations with branches cut-off during the exploration. Branching consists of eliminating those combinations that are most likely leading to a bad solution. The branching method would surely affect the quality of the solution and the computational time. A too rigid selection would risk eliminating combinations that lead to a good solution. On the contrary, a loose selection, although it will more likely contain an optimal solution, would require too much computational time. A simple scheme of Branch&Bound method is shown in Figure 4.2. The branches can be cut both because the solution has exceeded a certain limit value established at each iteration and both because the solution does not reflect some basic costraints of the mission (such as mission time or maximum propellant).

4.2.3 Evolutionary algorithms

Evolutionary algorithms uses mechanisms inspired by biological evolution, such as reproduction, mutation, recombination, and selection. As in implicit algorithms, the quality of a solution is

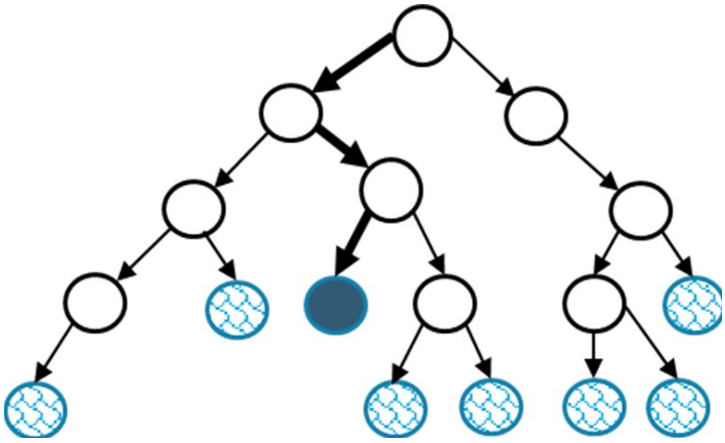


Figure 4.2: Diagram explaining the main concept of Branch&Bound algorithm.

evaluated through a fitness function that will guide the search towards an optimal solution. The most known evolutionary algorithms are *Genetic Algorithm* (GA) and *Ant Colony Optimization* (ACO).
GA is inspired by Charles Darwin's theory of natural evolution. This algorithm reflects the process of natural selection where the fittest individuals are selected for reproduction in order to produce offspring of the next generation. In the case of TSP, each chromosome, or individual, represent a possible debris sequence, and each debris of the sequence represents a gene. GA consists in five phases:

- Initial population
- Fitness function
- Selection
- Crossover
- Mutation

In the first phase a random population between all the possible combinations is chosen. Then, in the second phase the quality of the chromosomes is determined through a fitness function. Then, the fittest chromosomes are chosen in order to let their genes to be passed to next generations. In order to do that, two pairs of individuals (parents) are chosen based on their fitness score. Subsequently, selected parents will run into crossover. Depending on the crossover methods, like one-point, two-point or uniform crossover, two parents will mix their genes into two new individuals that will contain the same number of genes of parents. An example of crossovers with different methods is shown in 4.3. Moreover, in certain new offspring formed, some of their genes can be subjected to a mutation with a low random probability, and genes will be flipped forming a mutated individual. All the new chromosomes will form the new generation. This new generation will run into the same phases, and this loop will form all the next generations. When a satisfactory solution, i.e. the fittest individual, will be obtained, the code will stop.
ACO, as suggested by the name, is an algorithm based on the behaviour of ants of some species. In fact, they initially walk randomly, and upon finding food return to their colony while laying down pheromone trails. The ants will then be more likely to follow the paths already traced by the previous ants, i.e. the pheromone trail, to find more food. However, the pheromone trail tend to evaporate with time, and longer path will be penalized, leading the ants to look for other paths once the trail has completely evaporated. Transposing that to the optimization problem, the pheromone trail is the equivalent of the fitness function, indicating how good a solution is. Then, the algorithm will mainly follow the path with a larger pheromone trail,

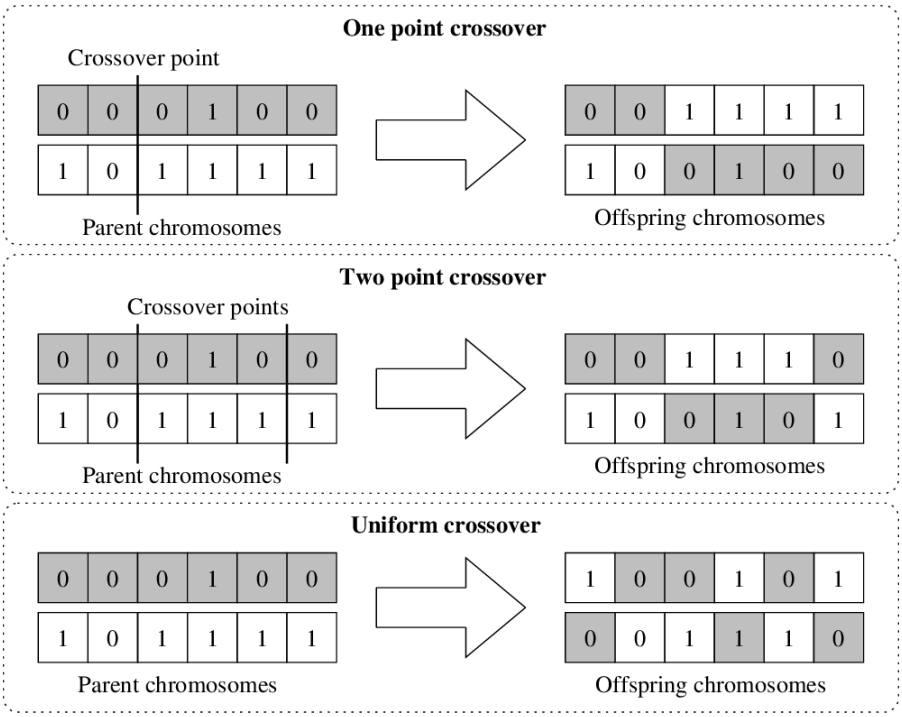


Figure 4.3: Illustration of one-point, two-point, and uniform crossover methods (from [37]).

without ceasing to look for other solutions in case the main trace starts to evaporate, i.e. lead to less optimal solutions. A concept scheme of ACO is shown in 4.4.
Evolutionary algorithms are particularly suited for large-sized NP problems.

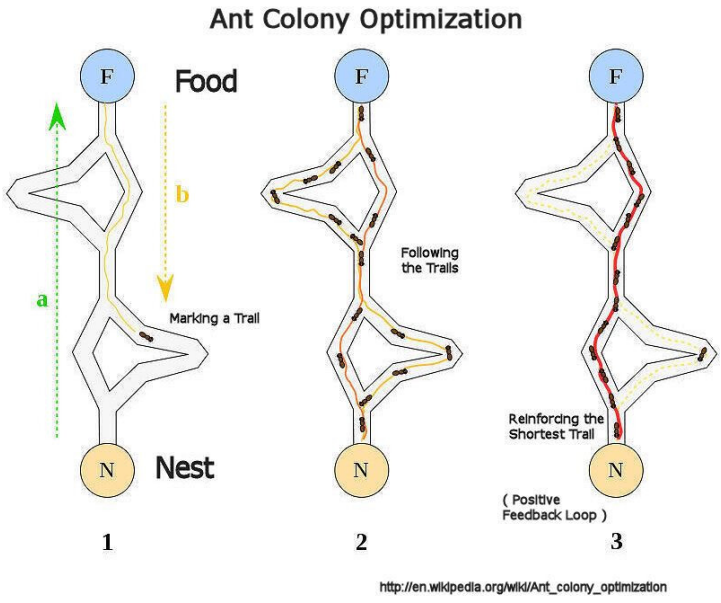


Figure 4.4: Concept scheme of Ant Colony Optimization (from [42]).

4.3 Cost function indices

In this dissertation, the quality of a solution is calculated with the help of a fitness function. This function will take into account not only the Δv required for transfers between debris,

but will also take into account some factors related to the impact of the debris on the orbital environment and the difficulty in capturing the debris itself. These two aspects will be taken into account, respectively, in the *Environmental Index* and in the *Operational Index*. These two indices have already been used in the paper of Borelli et al. [5]. They will be discussed in the next two paragraphs.

4.3.1 Environmental index

As anticipated, the environmental index describe the criticality of an inactive object to the debris environment. This index therefore expresses how convenient it would be to remove a certain debris to reduce the risk of collisions in that orbital region. Other authors have previously tried to quantify the impact of debris on the orbital environment. Liou and Johnson [25] consider the product between the mass and the collision probability, which depends on the spatial density and the relative velocities. Pardini and Anselmo [34] proposed a "Volumetric collision rate index", developed starting from analytical equations expressing the collision rate as a function of the fluxes of intact objects and cataloged debris pieces. Rossi et al. [36] defined a Criticality of Spacecraft Index (CSI), in which he considered the spatial density of the debris, the orbital lifetime, the mass and the inclination factor.
Inspired by these previous works, Borelli et al. defined the environmental index in this way:

$$I_{ENV} = \left(\frac{\Phi}{\Phi_0}\right) \cdot \left(\frac{M}{M_0}\right)^{1.75} \cdot \left(\frac{life}{life_0}\right) \tag{4.1}$$

where Φ is the flux of debris, M the debris mass and $life$ is the orbital lifetime function. The flux of debris, as described in Section 3.2, is calculated using the MASTER-8 simulation environment [16] and its value depends exclusively on the inclination and semi-major axis of the debris orbit. In fact, it gives indications on the flux of objects, expressed in kg/m^2, on a particular orbit for different object's size. In this dissertation, objects with sizes greater than 10 cm are considered. MASTER-8 will return a grid of flux values for given values of inclination and semi-major axis. The Matlab function *debris_flux.m*, taking as input the real values of i and a of the debris, will approximate them to the values closest to those of the data grid, returning the flux value for the approximated orbit.
The orbital lifetime function is an estimate of the time required for an inactive object in orbit to re-enter the atmosphere due to orbital perturbations, and in particular due to atmospheric drag. Thus, the lower the altitude of the debris, the faster it will re-enter the Earth's atmosphere. In fact, the drag force can be computed as

$$F_D = \frac{1}{2} C_D \cdot A \cdot \rho \cdot v^2 \tag{4.2}$$

where C_D is the drag coefficient, that will depends on object shape, A is the cross sectional area, ρ is the atmospheric density and v the object velocity. Atmospheric density will in turn depend on altitude, latitude and longitude, season, time of day, and solar and geomagnetic activities. Various more or less approximate models for estimating its value can be found in the literature. Similarly, the orbital lifetime can also be calculated both through simulations using semi-analytical propagators, like the Semi-analytic Tool for End of Life Analysis software (STELA) [17] designed by CNES, and with approximate formulas based on forecasting models. In this dissertation the formula adopted in McKnight et al. paper [30] has been used. Then, the lifetime is computed as

$$lifetime\left(h, \frac{S}{M}, \overline{F_{10.7}}\right) = f\left(\frac{S}{M}\right) \cdot g\left(\overline{F_{10.7}}\right) \cdot e^{ah^b + c} \tag{4.3}$$

where h is the altitude, S the cross sectional area, M the mass and $\overline{F_{10.7}}$ is the 10.7 cm solar flux, and where

$$f(x) = \frac{1}{\left[\left(\frac{x}{S_0/M_0} + 5\right)^{0.35} - 5^{0.35}\right] e^{\frac{x}{50\left(\frac{S_0}{M_0}\right)}}} \tag{4.4}$$

with $S_0/M_0 = 0.012 m^2/kg$, and

$$g(x) = \frac{116 \cdot 10^6}{x^{4.4}} \tag{4.5}$$

The value of the constants a, b and c in Equation 4.1 are taken from Rossi et al. [36] paper, and they are $a = 14.18$, $b = 0.1831$ and $c = -42.94$. To avoid excessively high lifetime values, a maximum value of 2000 years has been set. In Figure 4.5 it is shown how altitude and area-to-mass ratio can affect the lifetime of an inactive objects in LEO. It can be seen how its value increases with increasing altitude and decreasing S/M.

The 10.7 cm solar flux measurement is a determination of the strength of solar radio emission

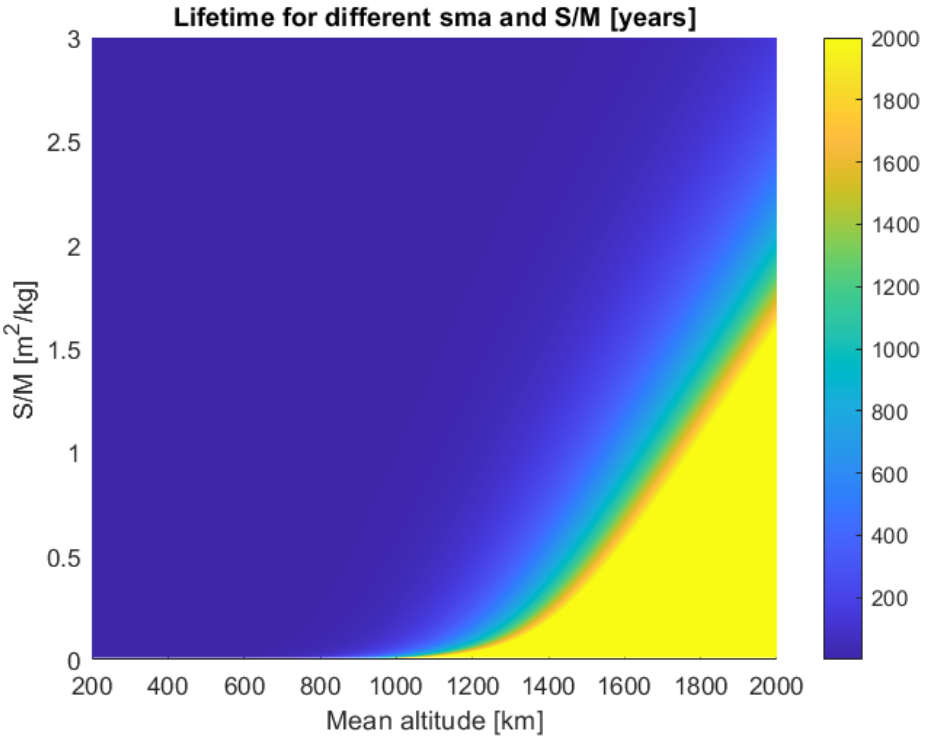


Figure 4.5: Lifetime values with varying values of altitude and area-to-mass ratio for objects in LEO, with maximum lifetime value of 2000 years.

in a 100 MHz-wideband centered on 2800 MHz (a wavelength of 10.7 cm), averaged over an hour. It is expressed in solar flux units (sfu), where $1 sfu = 10^{-22} W m^{-2} Hz^{-1}$. It is, along with sunspot number, one of the most widely used indices of solar activity. It has large variations from year to year, and these variations are mainly measured at the Penticton Radio Observatory in British Columbia, Canada. The data is provided by the National Research Council of Canada [7] and can be accessed from sites such as Space Weather Canada [8]. In Figure 4.6 can be observed the variation of the flux over the years.

I_{ENV} is then normalised with the value corresponding to an object of 1000 kg of mass in an orbit of 800 km of altitude and 98.5° of inclination.

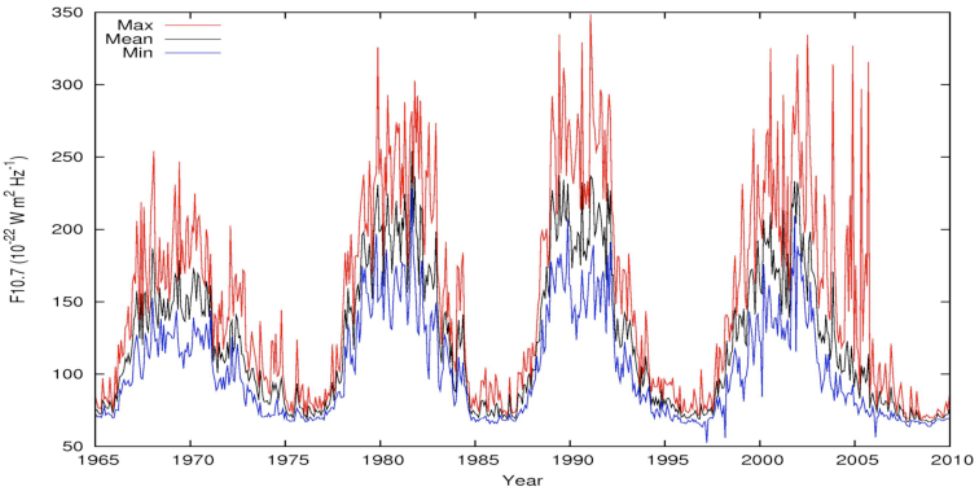


Figure 4.6: Mean, maximum and minimum solar radio flux F 10.7 in each 27-day solar rotation from 1965 to 2010 in solar radio flux unit (from [44]).

4.3.2 Operability index

This index quantifies the difficulties in approaching and capturing the debris. For the purposes of this thesis a rigid contact capture technique has been considered, for example through one or more robotic arms. In fact, it is a good technology in an ADR mission given the repeatability of the capture action, and it is also already widely used and therefore mature. The operability index takes into account three factors that will have a big influence on proximity operations:

- Attitude state
- Mass
- Illumination conditions

The *attitude state* represents the rotational state of the debris. The most important information for this purpose is the angular speed of rotation of the debris. As explained in Section 3.3.1, the angular velocity can be estimated from the light curves of the orbiting objects. An inactive object in orbit can be *stable*, if it does not have a periodicity in the light curve, or *rotating*, if it has a periodicity. In order to rigidly attach the debris, the chaser has to synchronize to its motion. Therefore, fast rotating objects will introduce more complications in the capture method.
Mass represents another major constraint since greater mass leads to greater catching difficulty. In fact, in case of rigid contact, rotating debris with a large mass can lead to the breakage of the robotic arms and therefore to the failure of the entire mission.
Finally, the *illumination conditions* are very important for proximity operations. In fact, once the chaser has arrived within a few tens of kilometers of the debris, a whole series of approach and docking operations will begin. For such operations, the chaser will mainly make use of sensors to precisely locate the satellite and to derive its precise shape and true attitude state. In order for these sensors to work, they need the target to be illuminated by the sun. For this reason, the debris whose orbit will be exposed to the sun for the longest time will be the favourites.
The operational index is therefore defined as follows:

$$I_{OP} = P_{ill} \left(\frac{a_{s0}(L, \omega_f)}{a_s(L, \omega_f)}) \right) \left(\frac{M_0 - M}{M_0} \right) \tag{4.6}$$

where P_{ill} is the percentage of the orbit with favorable lighting conditions, $a_s(L, \omega_f)$ is the estimated acceleration to obtain a full synchronisation, ω_f is the angular velocity derived from the apparent angular velocity and M is the mass of the debris. $a_s(L, \omega_f)$ is computed as

$$a_s(L, \omega_f) = L\omega_f^2 \qquad \text{with } \omega_f = \frac{\omega_a}{3°/s} \tag{4.7}$$

where L is the debris maximum length and ω_a is the apparent angular velocity. The formula has been obtained simply thinking of having to compensate the centrifugal acceleration in the reference system of the rotating debris.
I_{OP} is then normalised with the value of a debris with $M = 1000\ kg$, $\omega = 3°/s$ and $L = 2\ m$.

4.4 Branch&Bound algorithm

The algorithm developed in this dissertation is based on Branch&Bound algorithm concept. In fact, the search for an optimal solution will use a fitness function to quantify the quality of the solution and whether to proceed in that direction for the search, or to truncate that branch instead. As anticipated, the fitness function used for the algorithm will not only take into account the transfer cost required to reach the next detritus, but through the use of the two indices, I_{OP} and I_{ENV}, it will also take into account respectively how convenient it is to deorbit a certain debris based on the difficulty of capturing it and the benefit of deorbit it. The cost function will then consist of a weighted sum of these three elements. Since the goal of the algorithm will be to minimize it, the inverses of the two indices will be considered in the function. In fact, they quantify proportionally how convenient it is to deorbit a debris, and the aim is thus to maximize them. The cost function at each iteration will thus be computed as follows:

$$f_{COST} = \alpha_1 \cdot \frac{\Delta V}{\Delta V_0} + \alpha_2 \cdot \frac{I_{ENV}}{I_{ENV,0}} + \alpha_3 \cdot \frac{I_{OP}}{I_{OP,0}} \tag{4.8}$$

where α_1, α_2 and α_3 are the weights assigned respectively to the transfer cost ΔV, to the Environmental Index I_{ENV} and to the Operational Index I_{OP}. The three weights can be tuned according to mission requirements. Different mission scenarios with different weights will be presented Chapter 6.
First of all data have to be loaded. Debris data are collected into Matlab structures from which they can be easily extrapolated. As explained in Section 3.1, the population is taken from DISCOS database considering payloads and rocket bodies. These data are then organized in a more accessible matrix by the user defined function *data_ matrix.m*, considering only the derbis in a certain bin of inclination and semi-major axis chosen by the user. This matrix will contain the id of the debris and the values of the semi-major axis a, the inclination i, the eccentricity e, the right ascension of the ascending angle Ω, the mean anomaly M, the angle of perigee ω, the mass m, the apparent angular rate ω_a, the debris maximum dimension l_{max}, the percentage of the orbit in optimal illumination conditions P_{ill} and the cross-sectional area S. The matrix is organized like Table 5.7 These data are then used to create a precomputed matrix containing the Environmental Index and the Operational Index of all debris. First of all, the function *debris_ lifetime.m* uses S, m and a to compute the debris lifetime using Equations

n	id	a	i	e	Ω	M	ω	m	ω_a	l_{max}	P_{ill}	S
1	...											

Table 4.1: Example of the structure of the matrix variable *data* containing the information regarding all the debris analysed by the algorithm

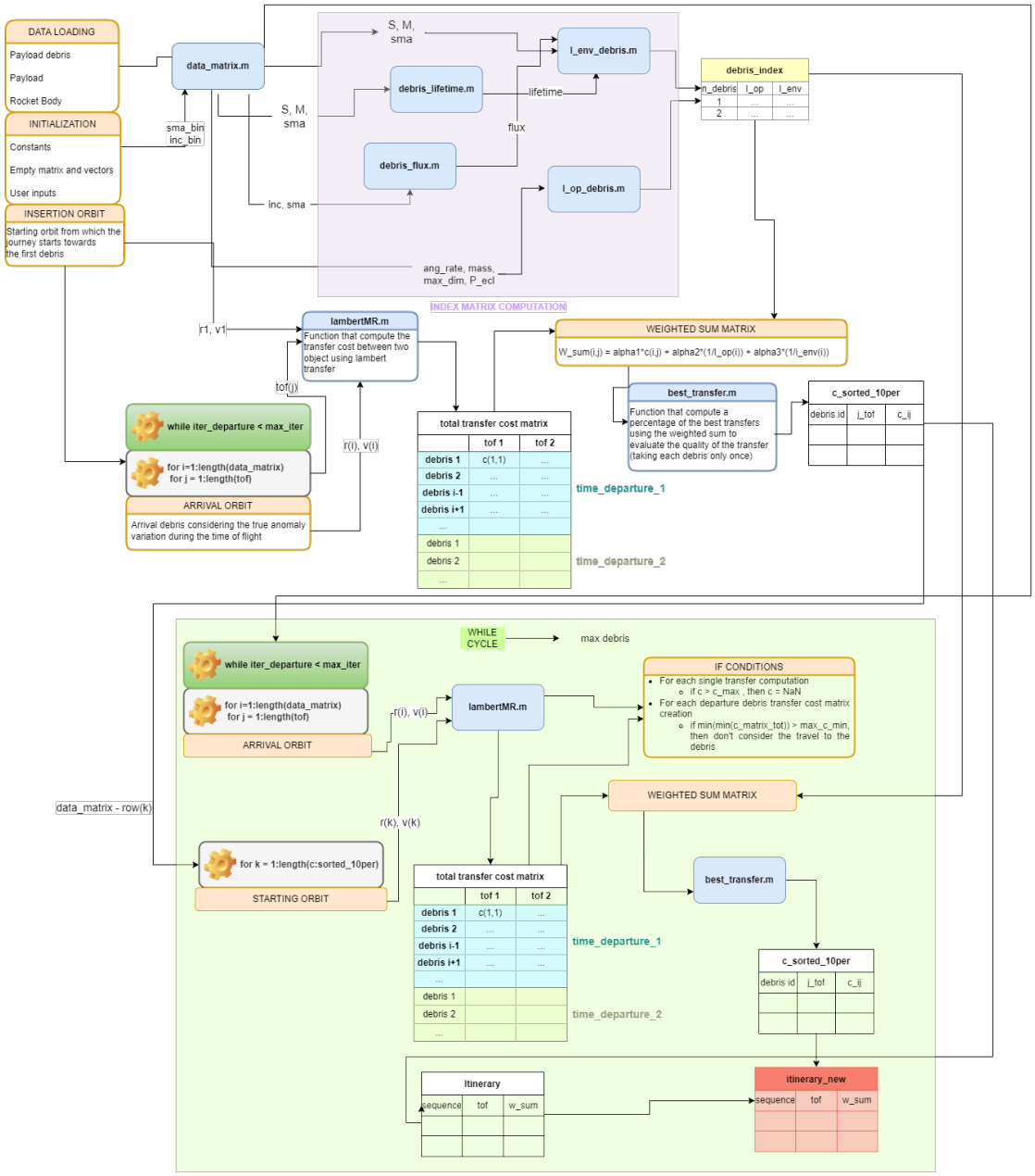


Figure 4.7: Block diagram of the algorithm which highlights the functions and their respective inputs and outputs.

4.3, 4.4 and 4.5. Then, i and a are used to compute the debris flux of the debris orbit by the function *debris_flux.m*. The lifetime and the flux will be then taken in input, together with S, m and a, by the function *I_env_debris.m* in order to compute I_{ENV} as in Equation 4.1. Similarly, the function *I_op_debris.m* take in input ω_a, m, P_{ill} and l_{max} in order to compute I_{OP} using Equations 4.6 and 4.7. The two indices for each debris are then collected into the matrix *debris_index*.

At this point, the iteration algorithm begins. The chaser is considered to start from an insertion orbit chosen by the user, that will have to specify its six orbital parameters. In first iteration, the algorithm will compute the transfer cost to reach all the considered debris. In order to consider more cases that can lead to more optimal solutions, this operation is repeated for 5 different departure times equally spaced. For each travel, the transfer cost Δv is computed using the function *lambertMR.m*. This function solve the Lambert's problem given the initial

and final position and velocity vectors and the desired time of flight, giving as outputs the initial and final velocity vectors of the Lambert transfer. This function can compute the transfer also for more revolution when possible, but for the aim of this dissertation the zero revolution case is considered. In order to find the more favorable transfer, the transfer cost is computed for different time of flight. The ToF vector is obtained by dividing the revolution time of the arrival orbit equally and considering only the times from the fourth element onward, since times that are too low would lead to solutions that are impossible or at any rate certainly too expensive. The position and the velocity vectors of the arrival debris are derived from its orbital parameters. Due to orbital perturbations, the mean anomaly will not be the only parameter to vary. In this dissertation, the J_2 perturbations due to Earth oblateness are considered. In this simplified perturbation model, the only parameters that vary are the RAAN, the Argument of Perigee (AoP) and the mean anomaly M. Their values will change over time in that way:

$$\frac{d\Omega}{dt} = -\frac{3}{2}\left(\frac{r_E}{p}\right)^2 nJ_2 \cos i \tag{4.9}$$

$$\frac{d\omega}{dt} = \frac{3}{4}\left(\frac{r_E}{p}\right)^2 nJ_2(5\cos^2 i - 1) \tag{4.10}$$

$$\frac{dM}{dt} = n + \frac{3}{4}\left(\frac{r_E}{p}\right)^2 nJ_2\sqrt{1-e^2}(3\cos^2 i - 1) \tag{4.11}$$

where n and p are respectively the mean velocity and the semilatus rectum, and they are computed as:

$$n = \sqrt{\frac{\mu_E}{a^3}} \tag{4.12}$$

$$p = a(1-e^2) \tag{4.13}$$

Then, it is necessary to consider their variation both during the waiting time between the different departures time and during the ToF considered in Lambert's problem. Furthermore, if any of the transfer cost computed is higher than a maximum value set by the user, that is 1800 m/s for this dissertation, the value is automatically saved as NaN. At the end of all these iterations, the matrix of all the transfer cost is obtained. This matrix will have as many columns as the length of the tof vector, and as many rows as the number of debris considered multiplied by the number of departure times.
Using this matrix, the respective matrix of weighted sums according to Equation 4.8 will then be calculated. The precomputed matrix of indices corresponding to each debris will also be used for this purpose. This matrix will then be taken as input by the function *best_ transfer.m*. This function sorts the transfers by the value of their weighted sum, considering each debris only once (thus avoiding considering the same debris for different tof and departure times). After sorting them, it considers only a percentage of the best transfers (determined by the user) while discarding all others. This will avoid having to consider too many combinations in subsequent iterations, leading the algorithm to have prohibitive computation times. This step is based on the assumption that good single transfers will probably lead to an optimal overall solution.
At this point the most advantageous debris to deorbit will then be added to the sequences stored in the Matlab cell array called *itinerary*. Here in fact at each iteration the best sequences will be stored, updating them from time to time with the debris coming out of the *best_ transfer.m* function. So at the first iteration we will have sequences of only one debris, at the second iteration of two debris, and so on, until we reach the desired number of debris. For each sequence, the total weighted sum of its component debris, the total tof, the indices of all debris, and the total Δv of transfers will also be stored.
Once the *itinerary* matrix is built at the end of the iteration, the next for loop will calculate the best transfers from the last debris of the sequences stored in *itinerary*. Since more and

more combinations will have to be considered as the length of the sequence increases, at each iteration the percentage of the best transfers that the *best_ transfer.m* function has to consider will be gradually reduced. In this dissertation, for example, we started from considering the best 10% at the first iteration to considering the best 2% at the last iteration.
At the end of the cycle of iterations, once the desired number of debris to be deorbited has been reached, a cell array *itinerary* containing all combinations of debris to be orbited deemed good will be obtained. At this point they will be reordered to best to worst in terms of total weighted sum and the best 15 will be taken and saved in a structure called *travels*. This variable will actually contain the best solutions found by the algorithm, which would then be the starting point for a more in-depth analysis for an ADR mission.

5. Simulation and results

This section will show the results for different mission scenarios. First, the user-selectable inputs common to all scenarios will be made explicit. Then, for each scenario the respective parameters used will be specified and the results will be presented and commented on.
First of all, the constants used in the code are specified, namely Earth's radius which is assigned an average value $r_T = 6378km$ and Earth's planetary gravitational constant with value $\mu_T = 398600.4418km^3/s^2$. Another variable that can be chosen is the number of maximum revolutions for the Lambert transfer. It is an input to the function *lambertMR.m*, which gives the possibility of multi-revolutionary transfers. Its value for all cases is set to 0, so only direct transfers will be considered in this dissertation. Other parameters to be set are the maximum Δv for each transfer, a value above which transfers to a certain detritus are not even considered. This value is by default $c_{MAX} = 1800$. Finally, we will need to decide what weight to give to each coefficient in the weighted sum, that is, the coefficients α_1, α_2 and α_3 from Equation 4.8. The value of these coefficients will determine the various test cases with which the code was tested.

5.1 Test cases

5.1.1 Test case 1 - Δv only

In order to test the behavior of the algorithm, the first test cases (1,2 and 3) are scenarios in which the aim is to optimize only one parameter at a time.
The first test was done by optimizing only transfer cost needed to reach each debris. What will be expected will be precisely that it will choose sequences that are as convenient as possible in terms of transfer but will not take into account the impact of satellites on the orbital environment or their difficulty to capture. The coefficients of the weighted sum are then set to $\alpha_1 = 1$, $\alpha_2 = 0$ and $\alpha_3 = 0$.
In Table 5.1 are listed the first ten sequence chosen by the algorithm. As can be seen, the

Sequence	**Total Δv [m/s]**	**Total I_{OP} [-]**	**Total I_{ENV} [-]**	**I_{OP} vector [-]**	**I_{ENV} vector [-]**
[57,47,55]	642	18.2251	3.7893	[6.0561,6.1083,6.0607]	[1.2631,1.2631,1.2631]
[57,47,56]	666	18.2630	3.7893	[6.0561,6.1083,6.0986]	[1.2631,1.2631,1.2631]
[57,45,38]	708	18.5105	2.8775	[6.0561,6.1708,6.2836]	[1.2631,1.2631,0.3513]
[57,47,49]	718	18.2460	3.7893	[6.0561,6.1083,6.0816]	[1.2631,1.2631,1.2631]
[57,45,55]	720	18.2876	3.7893	[6.0561,6.1708,6.0607]	[1.2631,1.2631,1.2631]
[57,47,54]	777	18.2755	3.7893	[6.0561,6.1083,6.1111]	[1.2631,1.2631,1.2631]
[57,45,49]	780	18.3085	3.7893	[6.0561,6.1708,6.0816]	[1.2631,1.2631,1.2631]
[57,47,66]	795	18.2441	3.7893	[6.0561,6.1083,6.0797]	[1.2631,1.2631,1.2631]
[57,47,38]	860	18.4480	2.8775	[6.0561,6.1083,6.2836]	[1.2631,1.2631,0.3513]
[9,22,28]	860	7.5471	0.6753	[0.5442,6.4233,0.5796]	[0.3287,0.1710,0.1756]

Table 5.1: First 10 sequences chosen by the algorithm for test case 1

sequences are ordered in terms of Δv, while some very low index values appear. For example, debris number 9 and 28 that appears in the last sequence has both indices very low. In fact, they have a moderately high apparent angular velocity ($\omega_a > 3\ rad/s$) making it difficult to capture, and it is also in a belt where the debris flux is not very high (with about $a = 1415\ km$ for the first and $a = 1610\ km$ for the second and both with about $i = 50°$) and their removal would therefore not have much impact. One can make the exact same reflections for debris 38, in the third and the ninth sequences, which has a low environmental index since it too is in a sparsely populated belt.

5.1.2 Test case 2 - I_{OP} only

In order to continue the verification of the code, the other test will prioritize only the *operational index*.
As can be seen in Table 5.2, the sequences are ordered solely by their operational index omitting the necessary Δv and operational index. It is worth mentioning that in any case the code automatically excludes individual transfers with a $\Delta v > 1800\ m/s$, so there may be debris with an even lower operational index that is not considered. Placing a limit on the transfer cost makes it possible to consider missions even with high priority given to the two indices without going for completely infeasible sequences from a Δv point of view.

Sequence	**Total Δv [m/s]**	**Total I_{OP} [-]**	**Total I_{ENV} [-]**	**I_{OP} vector [-]**	**I_{ENV} vector [-]**
[32,1,72]	3480	27.6816	0.7762	[6.5066,9.4522,11.7228]	[0.0721,0.0270,0.6771]
[72,41,1]	4452	27.5709	1.2782	[11.7228,6.3959,9.4522]	[0.6771,0.5741,0.0270]
[72,5,1]	3356	27.4433	1.0452	[11.7228,6.2683,9.4522]	[0.6771,0.3411,0.0270]
[5,72,1]	4487	27.4433	1.0452	[6.2683,11.7228,9.4522]	[0.3411,0.6771,0.0270]
[72,53,1]	4088	27.3406	1.9672	[11.7228,6.1656,9.4522]	[0.6771,1.2631,0.0270]
[72,59,1]	4020	27.3237	1.9672	[11.7228,6.1487,9.4522]	[0.6771,1.2631,0.0270]
[51,1,72]	3490	27.3103	1.9672	[6.1353,9.4522,11.7228]	[1.2631,0.0270,0.6771]
[72,51,1]	4233	27.3103	1.9672	[11.7228,6.1353,9.4522]	[0.6771,1.2631,0.0270]
[72,65,1]	3022	27.2804	1.9672	[0.6771,1.2631,0.0270]	[0.6771,1.2631,0.0270]
[57,1,72]	2766	27.2311	1.9672	[6.0561,9.4522,11.7228]	[1.2631,0.0270,0.6771]

Table 5.2: First 10 sequences chosen by the algorithm for test case 2

5.1.3 Test case 3 - I_{ENV} only

For completeness, as mentioned above the algorithm was also tested by giving priority to the environmental index alone. In Table 5.3 it can be seen how the selected sequences are actually sorted only by the I_{ENV} of the debris.

Sequence	**Total Δv [m/s]**	**Total I_{OP} [-]**	**Total I_{ENV} [-]**	**I_{OP} vector [-]**	**I_{ENV} vector [-]**
[71,51,3]	3479	6.6421	7.4754	[0.5053,6.1353,0.0015]	[4.8844,1.2631,1.3279]
[71,52,3]	3769	6.5917	7.4754	[0.5053,6.0849,0.0015]	[4.8844,1.2631,1.3279]
[71,57,3]	3534	6.5629	7.4754	[0.5053,6.0561,0.0015]	[4.8844,1.2631,1.3279]
[71,58,3]	3651	6.5543	7.4754	[0.5053,6.0475,0.0015]	[4.8844,1.2631,1.3279]
[71,67,3]	3410	6.6324	7.4754	[0.5053,6.1256,0.0015]	[4.8844,1.2631,1.3279]
[71,68,3]	3787	6.6644	7.4754	[0.5053,6.1576,0.0015]	[4.8844,1.2631,1.3279]
[71,61,3]	4199	6.6065	7.4754	[0.5053,6.0997,0.0015]	[4.8844,1.2631,1.3279]
[71,62,3]	4218	6.5940	7.4754	[0.5053,6.0879,0.0015]	[4.8844,1.2631,1.3279]
[71,63,3]	4440	6.6480	7.4754	[0.5053,6.1412,0.0015]	[4.8844,1.2631,1.3279]
[71,65,3]	3960	6.6122	7.4754	[0.5053,6.1054,0.0015]	[4.8844,1.2631,1.3279]

Table 5.3: First 10 sequences chosen by the algorithm for test case 3

5.1.4 Test case 4 - Balanced mission

This test case goes to simulate a mission scenario where you want to consider all three contributions in a balanced way, setting $\alpha_1 = 1$, $\alpha_2 = 0.6$ and $\alpha_3 = 0.6$. Although Delta v continues to have a greater weight for obvious feasibility reasons, the two indices will also have a not inconsiderable weight. In this way, among the sequences that will require less propellant use, the algorithm will look for debris that will have a modest impact on the orbital environment and that will not present too high a capture problem.
As can be seen in Table 5.4, the total delta v is quite close to the case where we consider only its value. Although at first the values of the indices also seem similar, it can be seen that in contrast to the case where only Δv is considered they do not go down, but still maintaining an excellent total transfer cost they remain high in all sequences. In fact, it can be seen that the total Δv in this case increases faster, precisely to leave room for sequences containing debris with good values of I_{OP} and I_{ENV}. All the most favorable first sequences will start from the same debris since, besides being favorable to reach from the selected starting orbit from a Δv point of view, it is also a debris with excellent values of both indices. Therefore, this little variety in the results should not be surprising.

Sequence	**Total Δv [m/s]**	**Total I_{OP} [-]**	**Total I_{ENV} [-]**	**I_{OP} vector [-]**	**I_{ENV} vector [-]**
[57,47,55]	642	18.2251	3.7893	[6.0561,6.1083,6.0607]	[1.2631,1.2631,1.2631]
[57,47,56]	666	18.2630	3.7893	[6.0561,6.1083,6.0986]	[1.2631,1.2631,1.2631]
[57,47,49]	718	18.2460	3.7893	[6.0561,6.1083,6.0816]	[1.2631,1.2631,1.2631]
[57,45,55]	720	18.2876	3.7893	[6.0561,6.1708,6.0607]	[1.2631,1.2631,1.2631]
[57,47,54]	777	18.2755	3.7893	[6.0561,6.1083,6.1111]	[1.2631,1.2631,1.2631]
[57,45,49]	780	18.3085	3.7893	[6.0561,6.1708,6.0816]	[1.2631,1.2631,1.2631]
[57,47,66]	795	18.2441	3.7893	[6.0561,6.1083,6.0797]	[1.2631,1.2631,1.2631]
[57,45,56]	872	18.3255	3.7893	[6.0561,6.1708,6.0986]	[1.2631,1.2631,1.2631]
[57,46,55]	925	18.2817	3.7893	[6.0561,6.1649,6.0607]	[1.2631,1.2631,1.2631]
[57,45,54]	964	18.3380	3.7893	[6.0561,6.1708,6.1111]	[1.2631,1.2631,1.2631]

Table 5.4: First 10 sequences chosen by the algorithm for test case 4

5.1.5 Test case 5 - I_{OP} and I_{ENV} priority

An additional test case considered is a scenario in which a decision is made to give much higher priority to indices than to transfer cost. To do this, the weights were assigned the value $\alpha_1 = 0.3$, $\alpha_2 = 1.5$ and $\alpha_3 = 1.5$.
As can be seen in Table 5.5, despite the low weight of Δv there are still sequences that have a very good total transfer cost value, while for other sequences it is clear that the code has prioritized the two indices. Moreover, since the two indices have the same weight, it can be seen that if the sequence has a good total value of a certain index, that of the other index is somewhat less good.

5.1.6 Test case 6 - Only I_{OP} and I_{ENV}

As a final test case, it was thought to equally consider only the two indices and instead ignore the Δv term in the weighted sum. As can be seen in Table 5.6, the chosen sequences all have a fairly high transfer cost. However, the indices are alternately very good, that is, in the case where one is not, the other is. In a more varied situation the indices should still both be good, however having considered few debris and with many values assumed and therefore equal to each other we will have many indices equal to each other especially with regard to the environmental index. Thus the behavior of the algorithm is fully justified.

Sequence	Total Δv [m/s]	Total I_{OP} [-]	Total I_{ENV} [-]	I_{OP} vector [-]	I_{ENV} vector [-]
[71,67,57]	1587	12.6870	7.4106	[0.5053,6.1256,6.0561]	[4.8844,1.2631,1.2631]
[71,67,46]	1673	12.7958	7.4106	[0.5053,6.1256,6.1649]	[4.8844,1.2631,1.2631]
[71,67,47]	1693	12.7392	7.4106	[0.5053,6.1256,6.1083]	[4.8844,1.2631,1.2631]
[57,47,55]	642	18.2251	3.7893	[6.0561,6.1083,6.0607]	[1.2631,1.2631,1.2631]
[71,67,58]	1710	12.6784	7.4106	[0.5053,6.1256,6.0475]	[4.8844,1.2631,1.2631]
[57,47,56]	666	18.2630	3.7893	[6.0561,6.1083,6.0986]	[1.2631,1.2631,1.2631]
[71,67,45]	1762	12.8017	7.4106	[0.5053,6.1256,6.1708]	[4.8844,1.2631,1.2631]
[57,47,49]	718	18.2460	3.7893	[6.0561,6.1083,6.0816]	[1.2631,1.2631,1.2631]
[57,45,55]	720	18.2876	3.7893	[6.0561,6.1708,6.0607]	[1.2631,1.2631,1.2631]
[71,63,45]	1812	12.8173	7.4106	[0.5053,6.1412,6.1708]	[4.8844,1.2631,1.2631]

Table 5.5: First 10 sequences chosen by the algorithm for test case 5

Sequence	Total Δv [m/s]	Total I_{OP} [-]	Total I_{ENV} [-]	I_{OP} vector [-]	I_{ENV} vector [-]
[71,64,45]	2513	12.8671	7.4106	[0.5053,6.1910,6.1708]	[4.8844,1.2631,1.2631]
[71,64,53]	2938	12.8619	7.4106	[0.5053,6.1910,6.1656]	[4.8844,1.2631,1.2631]
[71,53,64]	3851	12.8619	7.4106	[0.5053,6.1656,6.1910]	[4.8844,1.2631,1.2631]
[71,64,46]	2941	12.8612	7.4106	[0.5053,6.1910,6.1649]	[4.8844,1.2631,1.2631]
[71,68,64]	2546	12.8539	7.4106	[0.5053,6.1576,6.1910]	[4.8844,1.2631,1.2631]
[59,64,71]	3066	12.8450	7.4106	[6.1487,6.1910,0.5053]	[1.2631,1.2631,1.2631]
[71,59,64]	3403	12.8450	7.4106	[0.5053,6.1487,6.1910]	[4.8844,1.2631,1.2631]
[71,53,45]	4292	12.8417	7.4106	[0.5053,6.1656,6.1708]	[4.8844,1.2631,1.2631]
[71,64,63]	2900	12.8375	7.4106	[0.5053,6.1910,6.1412]	[4.8844,1.2631,1.2631]
[71,63,64]	2502	12.8375	7.4106	[0.5053,6.1412,6.1910]	[4.8844,1.2631,1.2631]

Table 5.6: First 10 sequences chosen by the algorithm for test case 6

5.2 Plots

This section will show some plots to help better understand the behavior of the algorithm. While before for each individual test case the chosen sequences were shown, to get a better overview it will be necessary to show how the contribution of each of the three parameters on sequence choice changes as the assigned weights and thus the test cases vary.
Once the different test cases have been established in the previous section, we are going to consider for each of them the total sum of Δv, I_{OP} and I_{ENV} for the first 20 sequences selected by the algorithm. These sums will then be normalized to the maximum value for each of the three parameters so that they can all be shown in the same bar graph. The result is the graph shown in Figure 5.1 As can be seen from the graph, the trend of values largely reflects the weights assigned to the 3 parameters in the different test cases. Recalling that the overall goal is to minimize Δv and maximize the two indices (since they are proportional to the quality of the solution), the bar graph shown can be analyzed.
It can be seen that in the first test case the value of Δv is the lowest among all the test cases, consistent with the fact that solutions are chosen solely on the basis of that. As expected, the values of the two indices are quite low since they are not taken into account at all. The same graph can be plotted by highlighting the three components in the weighted sum instead. In fact, since the goal of the algorithm is to minimize the cost function, and since instead the two indices are proportional to the quality of the debris, the inverses of the two indices will be considered in the weighted sum. Figure shows the bar graph with indeed the inverses of I_{OP} and I_{ENV} to show the different contributions to the weighted sum.
In test case 2 only I_{OP} is considered, which in fact will have its highest total value among all test cases. The other two parameters will have no contribution, and in fact will also have the maximum value of Δv and the minimum value of I_{ENV}.
Similarly, in the third test case there will be the maximum value of the environmental index, at the expense of the other two parameters that are not considered in the analysis.

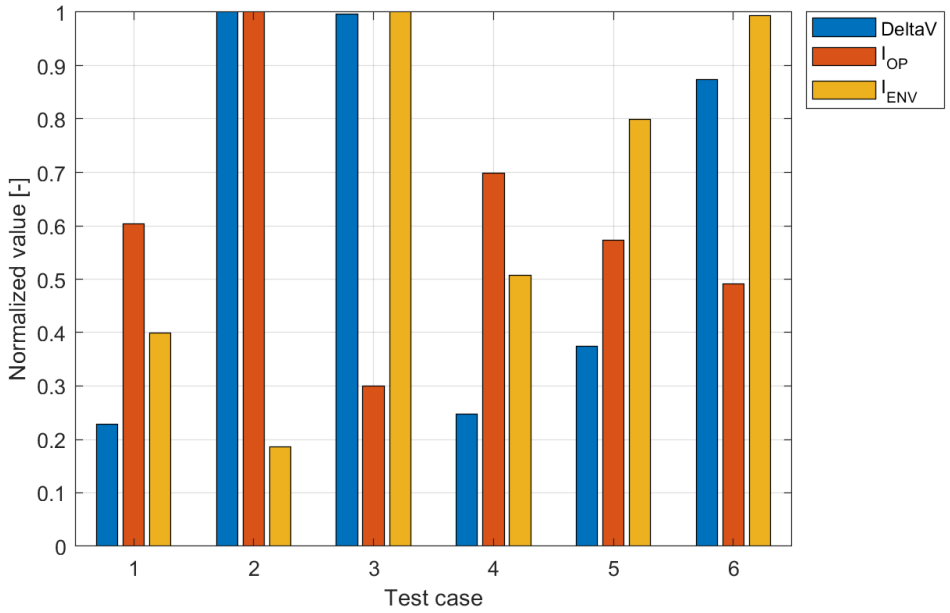


Figure 5.1: Bar graph of the normalized sum of Δv, I_{OP}, and I_{ENV} values for the first 20 sequences of each test case

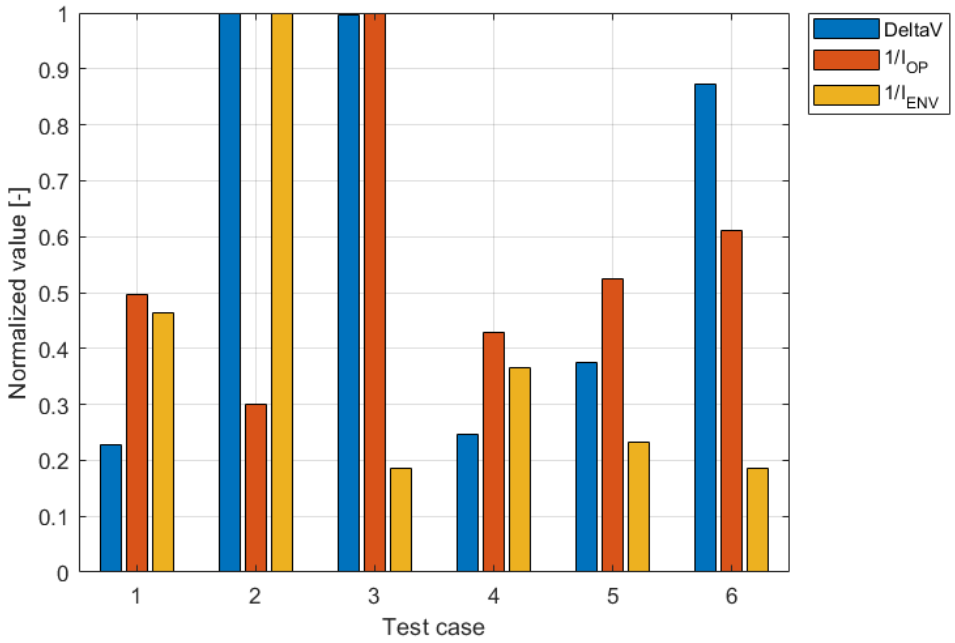


Figure 5.2: Bar graph of the normalized sum of the three contributions to the weighted sum Δv, $1/I_{OP}$, and $1/I_{ENV}$ that defines the functions cost that have to be minimized by the algorithm

In test case 4 we have a significantly more balanced situation. Although Δv has more weight, the two indices will also have a not insignificant contribution. In fact, although the total transfer cost is a little higher than in the case where only Δv was considered, the two indices will have quite larger values, thus leading to very good solutions from a collision risk and capture difficulty point of view.
In the fifth test case, compared with the previous one, it was decided to give much more

importance to the two indices, and this is also reflected in the bar graph. In fact, both indices will have a visibly higher total value, at the expense of the total transfer cost, which will also experience an increase.
In the final test case it is evident of how Δv was not taken into consideration given its high total value, while the two indices will have a significantly higher value than in the previous case. Regarding the latter, despite having the same weight, as mentioned earlier it can be seen how it still gave priority to the environmental index. As already explained this may be due to little variability in the data, or in any case it implies that it will simply need to assign a higher weight to the operational index in case it is to be given more importance.
As can be seen, the bar graph can be amply justified in each of the test cases considered and leads to results that might have been expected. This is certainly indicative of good behavior of the algorithm, and suggests that it would behave correctly even in scenarios other than the one considered.
Another interesting graph to show is that of the changes in RAAN of the different orbits and their difference during transfers. In fact, the direct change of RAAN is an extremely expensive maneuver, and what is best to do is to make transfers between two debris that have a small difference in RAAN at the time of transfer. Taking into account that the RAAN varies due to the oblateness of the Earth, differently depending on the inclination and semi-major axis, it will be expected that the code will prefer debris that also aligns as a consequence of the different rate of change of the RAAN. In Figure 5.3 it can be seen how the algorithm in the case where it needs to minimize only the Δv chooses debris that at the time of transfer has a similar RAAN and semi-major axis very close (if not equal). In Figure 5.4 it is shown the same plot for test

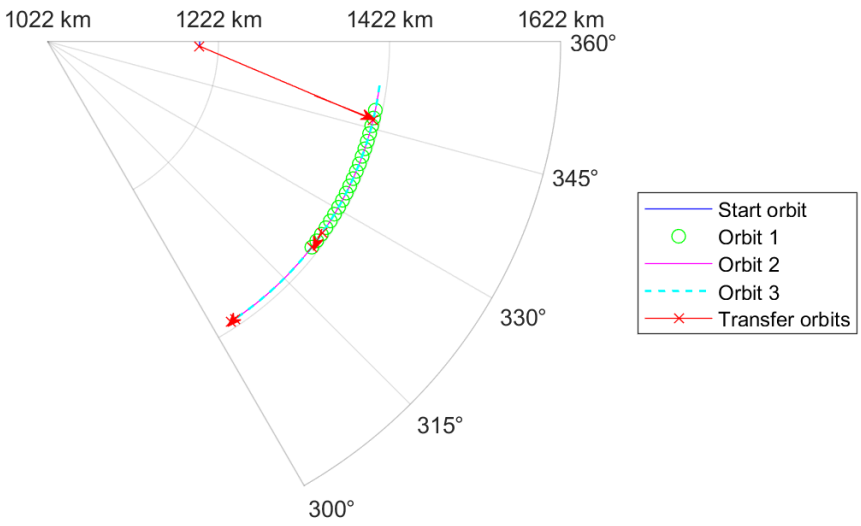


Figure 5.3: Polar plot of the RAAN change due to Earth's oblateness and sma and RAAN change due to transfers (in red) for test case 1

case 6, that is the opposite situation of test case 1. In fact the objective is to maximize the I_{OP} and the I_{ENV} of the debris to deorbit. In fact, it can be seen how the transfers cover a wider range in terms of sma and RAAN.However, since values of Δv that are too high are excluded by default anyway, the transfers chosen by the algorithm do not have prohibitive values anyway, and thus the differences in terms of RAAN are still very limited.

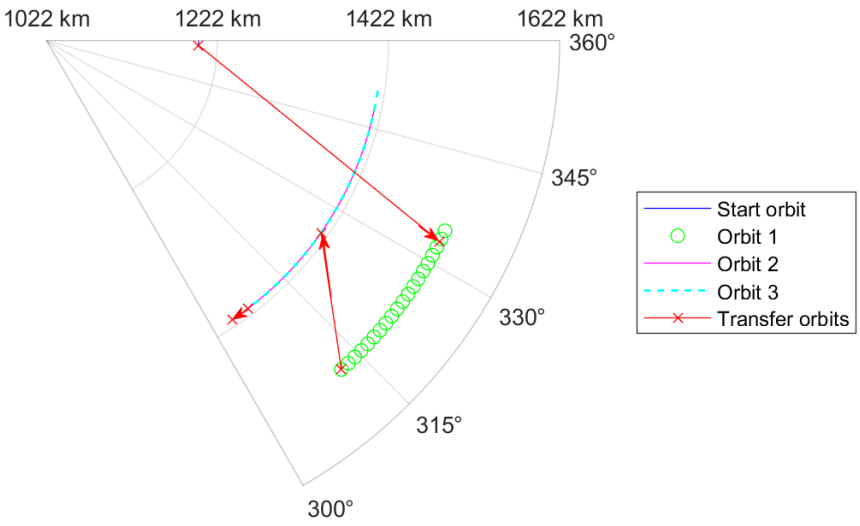


Figure 5.4: Polar plot of the RAAN change due to Earth's oblateness and sma and RAAN change due to transfers (in red) for test case 6

5.3 Data matrix

In order to better interpret the results, the data table regarding the debris population considered in the analysis has been provided in this section in Table 5.7.
Going to look at the different test cases, it can be seen that detritus number 71 figures as the one with the highest environmental index. With the data in hand, it is easy to see that this is mainly due to its significant mass, which would go to increase the risk of collision if it remained in orbit. However, one can also see how it has a very low operational index instead. In fact, its mass will also go to affect the difficulty of capture. In addition to that, it can also be seen that it also has a greater rotational velocity and maximum size than most debris, elements that will further go to make its capture more difficult.
In contrast, detritus 72 that also appears frequently reports high I_{OP} but low I_{ENV}. In fact, it will have a much smaller rotational velocity and maximum size, going to make capture easier, while it will have enough mass to make it a high-risk object for impact, especially having a semi-major axis and inclination such that it is in a belt with high debris flow.
The same arguments can be made for all other debris in the sequences selected in the different test cases. This only further confirms the validity of the code.

debris n°	debris id	a [m]	i [°]	e [-]	Ω [°]	M [°]	ω [°]	m [kg]	ω_a [rad/s]	L_{MAX} [m]	P_{ill} [-]	S [m^2]
1	27769	7525926.287	52.0205	0.0062394	190.1528	200.9825	197.9544	100	3	0.912854863	0.7263	0.833304
2	6071	7463094.256	52.0335	0.1187165	247.115	48.0704	300.8664	495	3	1.746596691	0.697742818	3.0506
3	16905	7865735.441	50.0078	0.0011333	167.7316	277.9361	249.1202	678.89	149.5363923	2.15	0.708914729	4.6225
4	25159	7892628.163	52.0011	0.0001487	120.071	168.8213	338.8338	450	2.666666667	12	0.744573643	144
5	25160	7787451.938	52.0025	0.000265	5.1211	288.2001	219.7869	450	1	12	0.729297766	144
6	25162	8006340.98	51.9563	0.0008034	152.4845	266.2395	107.6969	450	4.280618312	12	0.74129047	144
7	25306	7901727.802	51.9985	0.0001968	331.4593	155.8889	234.5645	450	1	12	0.731121751	144
8	25618	7791162.103	52.0131	0.0010687	356.6442	108.5823	74.4424	450	28.38904025	12	0.724532604	144
9	25619	7793171.482	51.9947	0.0000836	359.1069	130.4439	40.5892	450	3.396226415	12	0.730369357	144
10	25620	8119692.989	51.9974	0.0010855	67.6739	139.8292	343.1545	450	40.96649316	12	0.74749202	144
11	25621	7916859.397	51.9964	0.0002104	182.6669	233.5215	139.9439	450	3.416432668	12	0.736867305	144
12	25646	8048028.007	52.0037	0.0002389	213.9724	329.7339	43.568	450	57.87756212	12	0.738987688	144
13	25647	8074124.632	52.0033	0.0002475	59.9379	162.7505	197.3247	450	1	12	0.751139991	144
14	25648	8017371.027	52.0045	0.000262	206.6319	279.3754	94.2242	450	1	12	0.748267214	144
15	25649	7791404.597	51.9982	0.0002014	127.7056	19.7815	127.3944	450	1	12	0.737619699	144
16	25673	7940368.766	52.0003	0.0001271	21.6261	118.7953	52.6372	450	24.98007589	12	0.745759234	144
17	25674	7997678.279	51.9879	0.0001229	70.7549	63.4623	103.7595	450	13.87945343	12	0.741085271	144
18	25675	8103144.914	51.9963	0.0005661	235.9706	120.9059	319.4394	450	28.13899968	12	0.751413589	144
19	25676	7961852.307	51.9998	0.0002938	96.6094	301.2987	71.3845	450	55.53319008	12	0.747925217	144
20	25872	7840467.034	51.9635	0.0064	62.4958	295.7929	64.9559	450	6.857142857	12	0.735932513	144
21	25904	8089804.757	51.9964	0.0001894	336.6955	208.2104	151.8833	450	31.78994616	12	0.757273142	144
22	25905	8026020.65	51.999	0.0008645	71.2583	290.5425	232.0731	450	1	12	0.747332421	144
23	25906	7969622.289	51.9945	0.0013381	23.2503	304.5664	55.6447	450	50.43539961	12	0.743205654	144
24	25907	7930149.835	52.0007	0.0006654	91.0293	126.6249	233.3991	450	1	12	0.741586867	144
25	25940	7950007.753	52.0002	0.0002882	50.3914	236.243	123.8693	450	36.71591084	12	0.744733242	144
26	25941	8059242.045	51.9993	0.0000737	115.6839	195.3067	177.5532	450	27.76844999	12	0.751048792	144
27	25942	8037820.079	51.9942	0.0000899	147.3161	331.5999	41.1951	450	23.07914911	12	0.750820793	144
28	25943	7988240.608	51.994	0.0001444	109.5006	29.6507	343.3635	450	3.31797235	12	0.742339261	144
29	25958	7791858.333	51.984	0.0002289	212.5582	69.7859	290.2762	450	5.135520685	12	0.731600547	144
30	25959	8152545.715	51.9852	0.0014025	156.3927	10.2082	2.7462	450	3.673469388	12	0.761536708	144
31	25960	8059645.302	51.9841	0.0002686	4.352	166.4152	4.581	450	1	12	0.73495212	144
32	25961	8164866.804	51.9812	0.000527	325.1409	229.2691	301.8273	450	1	12	0.757022344	144
33	26078	8121691.728	51.9952	0.0004023	296.8018	291.299	239.9306	450	8.309239033	12	0.752621979	144
34	26079	7870428.634	51.9843	0.0007303	277.6597	249.7284	142.9896	450	6.14334471	12	0.737619699	144
35	26080	7981164.582	52.0019	0.0003269	73.2748	242.8018	284.202	450	6.165118822	12	0.747081623	144
36	26081	8110918.719	52.005	0.0001668	48.6746	355.5676	4.5172	450	1	12	0.755129959	144
37	31568	7791733.072	51.9895	0.0000772	168.4543	327.2802	141.2572	450	3	12	0.731463748	144
38	31570	7791730.245	51.9929	0.0000372	124.9017	39.1782	108.6652	450	1	12	0.731076151	144
39	31571	7791734.002	52.0078	0.0000198	219.7409	276.0807	97.1986	450	1	12	0.723187415	144
40	31573	7791716.508	52.0046	0.0000873	129.1264	66.0137	72.2286	450	1	12	0.727747378	144
41	32260	7882165.61	51.9722	0.0000922	200.5907	331.2335	43.5013	450	1	12	0.744140447	144
42	32261	7791770.883	51.9755	0.00009	73.652	24.8671	111.4075	450	1	12	0.728476972	144
43	32262	7791731.928	51.9672	0.0000823	295.7045	233.5479	162.8026	450	1	12	0.719448244	144
44	32263	7789819.106	51.9879	0.0001441	249.1566	292.737	95.9161	450	1	12	0.735430917	144
45	37185	7791725.224	52.0032	0.0000231	82.463	51.5005	84.7365	715	1	12	0.738440492	144
46	37186	7791735.455	52.0008	0.0000409	81.882	64.639	77.6345	715	1	12	0.737733698	144
47	37187	7791705.352	52.0028	0.0000253	82.1232	133.8133	345.815	715	1	12	0.730962152	144
48	37188	7791726.109	52.0095	0.0000599	38.1679	105.4651	65.8969	715	1	12	0.732740538	144
49	37189	7791742.701	51.9925	0.0000123	124.4894	0.2836	141.5655	715	1	12	0.727770178	144
50	37190	7791722.66	52.008	0.0000532	37.4029	104.1498	66.8332	715	1	12	0.73253534	144
51	37736	7791736.689	51.9885	0.00007	348.1692	65.9857	105.8501	715	1	12	0.734199726	144
52	37737	7791739.014	51.9724	0.0000656	254.58	244.8765	132.0229	715	1	12	0.728157775	144
53	37738	7791743.578	51.9991	0.0000648	215.7064	330.5714	142.4919	715	1	12	0.737824897	144
54	37739	7791743.183	51.9852	0.000042	168.1248	235.1773	138.4782	715	1	12	0.73130415	144
55	37740	7791724.586	51.9947	0.0000321	170.1144	212.1035	161.0057	715	1	12	0.725262198	144
56	37741	7791742.977	51.9909	0.000038	124.3564	92.5027	44.2471	715	1	12	0.729799362	144
57	38037	7791742.771	52.0032	0.0000707	351.4489	105.7083	67.8415	715	1	12	0.724715002	144
58	38038	7791734.644	52.0085	0.0000666	352.4888	109.2104	61.3133	715	1	12	0.72368901	144
59	38039	7791748.504	52.0076	0.000107	307.6358	258.5905	116.3018	715	1	12	0.735795714	144
60	38040	7791718.014	52.0107	0.0000805	37.6774	112.9588	58.845	715	1	12	0.732033744	144
61	38041	7791734.792	51.9883	0.0000705	303.4816	59.0351	112.0108	715	1	12	0.729936161	144
62	38042	7791740.422	51.9899	0.0000532	304.1029	269.3763	261.9959	715	1	12	0.728522572	144
63	39069	7791751.01	52.0039	0.0000585	216.5325	204.2097	155.8797	715	1	12	0.734906521	144
64	39070	7791737.125	51.9806	0.0000649	211.7794	196.5107	163.5781	715	1	12	0.740857273	144
65	39071	7791736.751	51.9935	0.0000177	125.7235	234.864	125.2243	715	1	12	0.730620155	144
66	39072	7791732.035	52.0063	0.0000381	172.807	197.2121	162.8759	715	1	12	0.72754218	144
67	39073	7791727.068	51.9866	0.0001468	258.7367	240.6677	119.4336	715	1	12	0.733036936	144
68	39074	7791730.216	51.9792	0.0001169	256.4323	226.065	134.0313	715	1	12	0.736867305	144
69	57681	7473637.317	50.008	0.0006731	99.5592	78.8273	281.185	500	3	5	0.697628819	25
70	447	7490910.564	50.1315	0.0065658	324.2332	245.2887	286.6616	445.99	3	4.8	0.698130415	23.04
71	16907	7916243.543	50.0058	0.007262	334.5254	106.0908	64.5768	2378.59	3	13.53	0.74753762	183.0609
72	25771	7463356.89	51.5295	0.0375243	343.2268	174.2532	355.7923	919	1	5.9	0.705221158	34.81
73	25873	7254863.445	51.7624	0.0530718	215.7362	249.9679	115.7025	919	3	5.9	0.676105791	34.81

Table 5.7: Population of debris used for the test cases

6. Conclusions

6.1 Results discussion

In this thesis, an algorithm was developed for choosing the sequence of debris to be deorbit in a multi ADR mission. Compared to previously developed methods, which went for an optimization considering only the Δv or at most a bi-objective optimization considering time and Δv, in this dissertation it was aimed to enrich the research with the implementation of the two indices I_{OP} and I_{ENV}. While Masserini's work [29] focused more on comparing two mission architectures (chaser and deorbiting kits) and the performance of different optimization algorithms based solely on the transfer cost, here more emphasis was placed on defining a cost function that would go into multiple aspects. The advantage of a multi-index cost function is to be able to maximize the benefits obtained from an ADR mission by deorbiting the debris with the greatest impact on the orbital belt, and to minimize the risks due to the high difficulty of capturing some debris, all while finding a good trade-off with the amount of propellant required. Although the algorithm makes use of several assumptions and simplifications, it still seems to lead to satisfactory results. In fact, always keeping the total Δv fairly limited, it succeeds in finding sequences whose debris has characteristics such that it has a higher risk of collision but still does not present too high a difficulty of capture, risking leading to failure of the entire mission due to irreversible damage to the OTV.
Despite the limited population of debris considered and the lack of various data, it is nonetheless possible to admire from the graphs how effectively the algorithm responds well to the variation of the weights given to the different contributions of the weighted sum. Being that many values are assumed it is easy to see many indices equal to each other, especially the environmental index. Indeed the latter has a strong dependence on the orbital bin it is considered, and thus its limited variation is mostly related to the choice of population of interest in this work. Moreover, the assumed values of cross sectional area and mass for the objects with no data available further influence the limited I_{ENV} variability. The assumptions on the missing data have an impact also on the operational index. In fact, the rotational rate plays a key role on definition of the ease of capture of an object. However, there is limited data on the in-orbit object rotational rate and is therefore often assumed according to light curves data, as can be seen Section 3.3.1. Despite this assumption, the focus of this thesis was the algorithm behaviour in objects selections for ADR mission according to multiple ranking indices. Therefore, the modelling and analysis of the relative behaviour of the multiple indices was more of interest. The choice to draw from real data was made to demonstrate how it is possible to derive all the data necessary for calculating the best sequence from databases that already exist and are available to all. To test the code, however, it would also have been sufficient to create an entire fictitious population of debris, so that there is more variability from the data. In fact, having values of the two indices all different from each other would allow a better view of the behavior of the algorithm as the assigned weights change.

6.2 Future works

This dissertation can be more of a starting point for future developments. In fact, many simplifications are considered throughout the thesis.
Firstly, different mission architectures can be studied. In Chapter 2 it was seen how some more or less elaborate mission architectures were possible. The analyzed architecture is on the whole quite simple: high thrust propulsion, direct RAAN change, and deorbiting kit strategy were chosen. For example, adopting electric thruster would certainly reduce the total Δv needed thus being able to afford to go after debris with somewhat higher indices. On the other hand, this will increase the complexity of the modelling and design of a low-thrust dynamics and transfer. Adopting a RAAN drift strategy for RAAN change would be beneficial from a propellant point of view, being the plane change manoeuvres expensive to perform, as it is one of the most expensive maneuvers. Moreover, this will allow a more through exploration of the debris population for the selection of the ADR target expanding the achievable RAAN ranges. Finally, the disposal orbit strategy would eliminate some of the complications associated with having to attach ready-made kits to the debris. However, depending on the capture method, it could lead to even more complex dynamics such as in the use of a semi-rigid trapping method, like harpoons and nets, which has less predictable behavior than a rigid method, or in the use of methods that increase the aerodynamic drag of the debris, like sails, which must be studied according to its mass and altitude.
From the algorithm point of view, some improvements can be foreseen. The chosen method, inspired by Branch&Bound, is efficient for medium-sized problems. However, in more complex scenarios where the population of debris is significantly larger it an evolutionary algorithm may be more suitable. In fact, in these cases a Branch&Bound scheme will be less efficient from a computational effort point of view. In addition, having adopted the RAAN direct change strategy and high-thrust propulsion, it was not deemed necessary to consider the total mission time among the parameters of the cost function since all the solutions found had very short total times. However, a future implementation could be to extend the cost function formulation to consider also the total time of the ADR mission, often an important parameter for mission planning and design.
A further possible implementation is to consider the propellant required to bring a debris to a reentry orbit by simply using the *Tsiolkovsky rocket equation*. In this way, by setting a maximum amount of propellant with which each kit can be filled, one can eliminate those solutions that include debris that would require more propellant.
In essence, many implementations can and should be made especially for its application in a real mission scenario. What this dissertation aimed at was to demonstrate how it was possible to use an index-based satellite rating system in selecting an optimal sequence of debris to orbit. Thus, the use of other types of indices, as discussed in the paper of Borelli et al. [5], or alternative indices should not be ruled out.

Bibliography

[1] European Space Agency. *ClearSpace-1*. 2019. URL: https://www.esa.int/Space_Safety/ClearSpace-1.

[2] European Space Agency. *ESA's e.Deorbit debris removal mission reborn as servicing vehicle*. 2018. URL: https://www.esa.int/Space_Safety/Clean_Space/ESA_s_e.Deorbit_debris_removal_mission_reborn_as_servicing_vehicle.

[3] Richard H. Battin. "Solving Lambert's Problem". In: *An Introduction to the Mathematics and Methods of Astrodynamics, Revised Edition*. 1999, pp. 295–364.

[4] Nicolas Bérend and Xavier Olive. "Bi-objective optimization of a multiple-target active debris removal mission". In: *Acta Astronautica* 122 (May 2016), pp. 324–335. ISSN: 0094-5765. DOI: 10.1016/J.ACTAASTRO.2016.02.005.

[5] Giacomo Borelli et al. "A Comprehensive Ranking Framework For Active Debris Removal Mission Candidates". In: *8th European Conference on Space Debris*. Ed. by T. Flohrer, S. Lemmens, and F. Schmitz. ESA Space Debris Office, 2021.

[6] Vitali Braun et al. "Active debris removal of multiple priority targets". In: *Advances in Space Research* 51 (9 May 2013), pp. 1638–1648. ISSN: 0273-1177. DOI: 10.1016/J.ASR.2012.12.003.

[7] NRC Canada. "Solar Weather monitoring". In: URL: https://nrc.canada.ca/en/research-development/products-services/technical-advisory-services/solar-weather-monitoring (visited on 03/22/2023).

[8] Space Weather Canada. "Solar radio flux". In: URL: https://www.spaceweather.gc.ca/forecast-prevision/solar-solaire/solarflux/sx-en.php (visited on 03/22/2023).

[9] Marilena Di Carlo, Juan Manuel Romero Martin, and Massimiliano Vasile. "Automatic Trajectory Planning for Low-Thrust Active Removal Mission in Low-Earth Orbit". In: *Advances in Space Research* 59 (Oct. 2016). DOI: 10.1016/j.asr.2016.11.033.

[10] Lorenzo Casalino and Dario Pastrone. "Active Debris Removal Missions with Multiple Targets". In: *AIAA/AAS Astrodynamics Specialist Conference*. 2014. DOI: 10.2514/6.2014-4226. eprint: https://arc.aiaa.org/doi/pdf/10.2514/6.2014-4226. URL: https://arc.aiaa.org/doi/abs/10.2514/6.2014-4226.

[11] M. Cerf. "Multiple Space Debris Collecting Mission—Debris Selection and Trajectory Optimization". In: *Journal of Optimization Theory and Applications* 156 (3 2013), pp. 761–796. ISSN: 1573-2878. DOI: 10.1007/s10957-012-0130-6. URL: https://doi.org/10.1007/s10957-012-0130-6.

[12] *Chemical Propulsion Systems*. URL: https://www1.grc.nasa.gov/research-and-engineering/chemical-propulsion-systems/.

[13] P Colmenarejo et al. "GNC Aspects for Active Debris Removal". In: *CEAS EuroGNC. Delft, The Netherlands* (2013).

[14] Howard D. Curtis. "Orbital maneuvers". In: *Orbital Mechanics for Engineering Students* (Jan. 2020), pp. 287–350. DOI: 10.1016/B978-0-08-102133-0.00006-4. URL: https://linkinghub.elsevier.com/retrieve/pii/B9780081021330000064.

[15] ESA. "DISCOSweb". In: URL: https://discosweb.esoc.esa.int/ (visited on 02/16/2023).

[16] ESA. "MASTER". In: URL: https://sdup.esoc.esa.int/master/ (visited on 06/14/2023).

[17] Jean-François Goester, Aurélie Bellucci, and Claire Fremeaux. "CNES tools and methods for de-orbit and re-entry trajectories evaluation". In: $9^{TH} European Conference For Aeronautics And Space$ (2021). DOI: 10.13009/EUCASS2022-7020.

[18] Simeng Huang. "Multi-phase mission analysis and design for satellite constellations with low-thrust propulsion". Supervisor: Colombo, C. PhD thesis. Politecnico di Milano, 2021. URL: http://hdl.handle.net/10589/169779.

[19] Smithsonian Institute. "The MMT Observatory". In: URL: https://www.mmto.org/# (visited on 02/16/2023).

[20] D. Izzo and M. Maertens. "The Kessler Run: On the Design of the GTOC9 Challenge". In: *Acta Futura* 11 (Jan. 2018), pp. 11–24. DOI: 10.5281/zenodo.1139022. URL: https://zenodo.org/record/1139022.

[21] Dr. T.S. Kelso. "Celestrak". In: URL: https://celestrak.org/ (visited on 02/16/2023).

[22] Donald J. Kessler and Burton G. Cour-Palais. "Collision frequency of artificial satellites: The creation of a debris belt". In: *Journal of Geophysical Research: Space Physics* 83.A6 (1978), pp. 2637–2646. DOI: https://doi.org/10.1029/JA083iA06p02637. URL: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/JA083iA06p02637.

[23] Philippe Lacomme, Christian Prins, and Marc Sevaux. "Algorithmes de Graphes". In: (Jan. 2003).

[24] J. C. Liou. "An active debris removal parametric study for LEO environment remediation". In: *Advances in Space Research* 47 (11 June 2011), pp. 1865–1876. ISSN: 0273-1177. DOI: 10.1016/J.ASR.2011.02.003.

[25] J. C. Liou and Nicholas L. Johnson. "A sensitivity study of the effectiveness of active debris removal in LEO". In: *Acta Astronautica* 64 (2-3 Jan. 2009), pp. 236–243. ISSN: 0094-5765. DOI: 10.1016/J.ACTAASTRO.2008.07.009.

[26] Yong Liu and Jianan Yang. "A Multi-Objective Planning Method for Multi-Debris Active Removal Mission in LEO". In: *AIAA Guidance, Navigation, and Control Conference* (2017). DOI: 10.2514/6.2017-1733. URL: https://arc.aiaa.org/doi/abs/10.2514/6.2017-1733.

[27] *Low thrust trajectory optimization*. URL: https://www.agi.com/missions/space-operations-missions/low-thrust-trajectory-optimization.

[28] Dalal Madakat, Jérôme Morio, and Daniel Vanderpooten. "Biobjective planning of an active debris removal mission". In: *Acta Astronautica* 84 (Mar. 2013), pp. 182–188. DOI: 10.1016/j.actaastro.2012.10.038.

[29] Alessandro Maria Masserini. "Design and optimisation of an active debris removal service for large satellite constellations". Supervisor: Camilla Colombo, Co-supervisor: Simeng Huang. Master thesis in Space Engineering. Politecnico di Milano, Faculty of Industrial Engineering, Department of Aerospace Science and Technologies, 2021.

[30] Darren McKnight et al. "Identifying the 50 statistically-most-concerning derelict objects in LEO". In: *Acta Astronautica* 181 (2021), pp. 282–291. ISSN: 0094-5765. DOI: https://doi.org/10.1016/j.actaastro.2021.01.021. URL: https://www.sciencedirect.com/science/article/pii/S0094576521000217.

[31] J. Murakami and Shinji Hokamoto. "Approach for optimal multi-rendezvous trajectory design for active debris removal". English. In: *61st International Astronautical Congress 2010, IAC 2010*. Vol. 7. 61st International Astronautical Congress 2010, IAC 2010 ; Conference date: 27-09-2010 Through 01-10-2010. 2010, pp. 6013–6018. ISBN: 9781617823688.

[32] *Orbital Drag Illustration*. URL: https://www.eurekalert.org/multimedia/726264.

[33] Christos H. Papadimitriou. "Computational Complexity". In: *Encyclopedia of Computer Science*. GBR: John Wiley and Sons Ltd., 2003, pp. 260–265. ISBN: 0470864125.

[34] Carmen Pardini and Luciano Anselmo. "Using the space debris flux to assess the criticality of the environment in low Earth orbit". In: *Acta Astronautica* 198 (Sept. 2022), pp. 756–760. ISSN: 0094-5765. DOI: 10.1016/J.ACTAASTRO.2022.05.045.

[35] Glenn E Peterson. "Target identification and Delta-V sizing for active debris removal and improved tracking campaigns". In: *Proceedings of the 23rd International Symposium on Spaceflight Dynamics, Pasadena, Paper No. ISSFD23-CRSD2-5*. Vol. 29. 2012.

[36] A. Rossi, G. B. Valsecchi, and E. M. Alessi. "The Criticality of Spacecraft Index". In: *Advances in Space Research* 56 (3 Aug. 2015), pp. 449–460. ISSN: 0273-1177. DOI: 10.1016/J.ASR.2015.02.027.

[37] Kumara Sastry and David Goldberg. "Analysis of Mixing in Genetic Algorithms: A Survey". In: (June 2002).

[38] Hong-Xin Shen et al. "Optimization of Active Debris Removal Missions with Multiple Targets". In: *Journal of Spacecraft and Rockets* 55 (Oct. 2017), pp. 1–9. DOI: 10.2514/1.A33883.

[39] Jiří Šilha et al. "Apparent rotation properties of space debris extracted from photometric measurements". In: *Advances in Space Research* 61 (3 Feb. 2018), pp. 844–861. ISSN: 0273-1177. DOI: 10.1016/J.ASR.2017.10.048.

[40] *Space Debris and Human Spacecraft*. URL: https://www.nasa.gov/mission_pages/station/news/orbital_debris.html.

[41] *Travelling Salesman Problem in Java*. URL: https://www.javatpoint.com/travelling-salesman-problem-in-java.

[42] Wikipedia. "Ant colony optimization algorithms". In: URL: https://en.wikipedia.org/wiki/Ant_colony_optimization_algorithms (visited on 03/15/2023).

[43] M Yakovlev. "The "IADC Space Debris Mitigation Guidelines" and supporting documents". In: *4th European Conference on Space Debris*. Vol. 587. ESA Publications Division Noordwijk, The Netherlands. 2005, pp. 591–597.

[44] Andrew Yau, William Peterson, and Takumi Abe. "Influences of the Ionosphere, Thermosphere and Magnetosphere on Ion Outflows". In: June 2011, pp. 283–314. ISBN: 978-94-007-0500-5. DOI: 10.1007/978-94-007-0501-2_16.