TESIS DOCTORAL

Mixed integer nonlinear programming applications for trajectory optimization of large-scale active debris removal missions in low Earth orbit

Autor:

Adrián Barea Vilar

Directores:

Luis Cadarso Morga
Hodei Urrutxua Cereijo

Programa de Doctorado en

Tecnologías de la información y las comunicaciones

Escuela Internacional de Doctorado

2023

I have learned throughout my life as a composer chiefly through my mistakes and pursuits of false assumptions, not by my exposure to founts of wisdom and knowledge.

— Igor Stravinsky

ABSTRACT

Upcoming active space debris removal missions will most likely attempt to remove several objects per mission. The design of such missions involves the selection of the objects to be removed, as well as the optimization of the visit sequence and the orbital transfers interconnecting them.

This thesis focuses on the efficient resolution of optimization problems that involve the aforementioned kind of missions. In particular, the considered candidate pools of objects have to be large enough to be representative of the distribution of the most hazardous objects in the region of interest. Thus providing a more realistic view of the actual deorbiting capabilities of multi-target missions.

The efficient resolution of such large-scale instances poses three particular challenges, namely, the combinatorial complexity resulting from the size of the candidate object pool, the optimization of the orbital maneuvers and the interaction between the object selection and the maneuver optimization.

The combinatorial complexity of the problems has been addressed with a Mixed Integer Linear Programming formulation that prevents the appearance of solutions with disjoint subtours.

Regarding the maneuver optimization, both impulsive and low-thrust transfers have been considered. For impulsive maneuvers, a general Nonlinear Programming model has been proposed. Moreover, a dual-based method that is able to efficiently solve specific instances of multi-impulse maneuvers, while guaranteeing the convergence and the global optimality of the solutions, has been devised. For low-thrust maneuvers, this work presents a methodology to compute J_2-perturbed low-thrust transfers between circular orbits that achieves an advantageous trade-off between the fidelity of the orbital dynamics, the optimality of the transfers and the computational efficiency.

The interaction between the combinatorial decisions and the orbital dynamics has been handled with a two-stage approach that encapsulates each component of the problem in a stage. Conversely, an integrated Mixed Integer Linear Programming model that seamlessly coordinates the maneuver optimization and object selection has also been proposed. Furthermore, a Constraint Programming framework has been devised to deal with general mission analysis problems.

RESUMEN

ANTECEDENTES

El ritmo de crecimiento del número de objetos en órbita alrededor de la Tierra es un tema que requiere una gran atención, especialmente en las regiones más pobladas, las órbitas bajas terrestres (LEO) y las órbitas geoestacionarias, ya que para una densidad de objetos lo suficientemente grande existe cierta probabilidad de que se produzca una cadena de colisiones que genere nuevos objetos los cuales, a su vez, aumenten la probabilidad de que se produzcan nuevas colisiones, fenómeno que limitaría gravemente las operaciones en dichas regiones.

Para mitigar el crecimiento descontrolado de la basura espacial, el Inter-Agency Space Debris Coordination Commitee publicó una serie de recomendaciones. Entre tales recomendaciones, se aconseja que los objetos en la región LEO hayan reentrado en la atmosfera a los 25 años después de la finalización de su vida útil. Sin embargo, se han llevado a cabo investigaciones que indican que la aplicación estricta de esta medida sería insuficiente para que se estabilizara el número de objetos en órbita LEO; por lo tanto, asignando un determinado valor de amenaza a los objetos en función de su masa y probabilidad de colisión, proponen la eliminación de un determinado número de objetos, con grandes valores de amenaza, cada año para estabilizar el número total de objetos en dicha región.

Este interés por la eliminación activa de basura espacial ha motivado la investigación sobre la optimización de trayectorias de misiones en las que un satélite efectúa visitas a una serie de objetos para eliminarlos, bien deorbitándolos, o bien reubicándolos en órbitas en las que no supongan una peligro.

A la vista de los artículos publicados a fecha de inicio de esta tesis sobre optimización de trayectorias para la eliminación de basura espacial, no se ha explorado satisfactoriamente la posibilidad de elegir entre un gran número de objetos a eliminar con características orbitales diversas, ni se ha abordado el problema con las herramientas de optimización más apropiadas ni las más novedosas disponibles, lo que proporciona una clara oportunidad de mejora en el estudio de este caso práctico. Por lo tanto, el estudio de este problema no solo resulta de gran interés, sino que ofrece margen de mejora respecto al actual estado del arte. Explorando ambas vías de mejora, se deberían lograr unos mejores algoritmos que permitan tratar de manera efectiva un conjunto de objetos suficientemente representativos como para poder diseñar misiones de eliminación activa de basura espa-

cial que supongan un impacto global en la estabilización del número de objetos en órbitas LEO, condicionado a los recursos de los que se disponga para la realización de dichas misiones.

OBJETIVOS

El objetivo de esta tesis es la resolución de problemas de optimización de trayectorias de satélites para llevar a cabo misiones de eliminación de basura espacial en órbitas bajas terrestres, tal que, los objetos a eliminar se obtengan a partir de un conjunto lo suficientemente grande como para ser representativo de la distribución de los objetos más peligrosos en la región de interés.

La resolución eficiente de este tipo de problemas presenta tres dificultades. La más evidente es la complejidad combinatoria que conlleva elegir entre un número grande de objetos. Por ello, una exploración completa del espacio de posibles secuencias de objetos requiere del uso de técnicas que reduzcan considerablemente el número de dichas secuencias a evaluar.

Por otro lado, la determinación de la factbilidad de cada una de las secuencias evaluadas implica la optimización de las maniobras a realizar en la misión. A su vez, dicha optimización evalúa repetidamente la mecánica orbital que modela el movimiento de cada uno de los objetos implicados, generando así una carga computacional considerable. A consecuencia de esto, es importante hacer uso de algoritmos de optimización que consigan un compromiso entre la fidelidad de la dinámica orbital, la optimalidad de las maniobras y la eficiencia computacional del método.

Finalmente, hay que tener en cuenta que la selección de objetos y la optimización de las maniobras involucran decisiones de naturaleza diametralmente opuesta. Ya que, mientras la selección de objetos requiere la determinación de un conjunto de variables discretas, la optimización de maniobras implica la determinación de un conjunto de variables continuas sujetas al modelo no lineal y no convexo descrito por la dinámica orbital. La resolución de problemas de optimización no convexa con variables discretas es, en general, complicado. Por esa razón, la interacción entre la selección de objetos y la optimización de maniobras es un reto a superar. Las metodologías propuestas en esta tesis tienen que ser capaces de lidiar con estas tres dificultades, de cara a poder resolver de forma eficiente los problemas de optimización mencionados anteriormente.

METODOLOGÍA

Los métodos propuestos en esta tesis se basan principalmente en la resolución de modelos de Programación Entera Mixta y Programación No Lineal, los cuales se han complementado con técnicas de

control óptimo y programación con restricciones, así como con métodos heurísticos y metaheurísticos. De esa forma, se han configurado algoritmos compuestos capaces de cumplir los objetivos propuestos anteriormente.

RESULTADOS

Se ha tratado la complejidad combinatoria de los problemas por medio de una formulación de Programación Entera Mixta que evita la aparición de ciclos disjuntos. Esto da lugar a relajaciones lineales muy ventajosas, las cuales permiten reducir considerablemente el número de posibles soluciones a evaluar.

Respecto a la optimización de maniobras, por un lado, se ha desarrollado una metodología que explota las propiedades de la dualidad para la optimización de casos particulares de maniobras impulsivas con múltiples impulsos. Por otro lado, se ha formulado una técnica para optimizar maniobras de bajo empuje, sometidas a la perturbación gravitatoria del armónico J2, para transferir satélites entre órbitas circulares. Esta técnica es capaz de conseguir un compromiso muy ventajoso entre la fidelidad de la dinámica orbital, la optimalidad de las maniobras y el tiempo computacional necesario para calcularlas.

Por último, se han definido dos formas de enfocar la interacción entre la selección de objetos y la optimización de maniobras. La primera de ellas consiste en usar dos etapas para encapsular cada una de las dos componentes del problema. La otra técnica unifica ambas componentes, consiguiendo así un único modelo de Programación Entera Mixta.

CONCLUSIONES

Se ha conseguido tratar de forma eficiente problemas de optimización de alta complejidad computacional relacionados con el diseño preliminar de misiones de eliminación de basura espacial, cubriendo así un hueco existente en la en la literatura. De esa forma, se han desarrollado técnicas capaces de abordar problemas de aplicación muy elaborados y de gran relevancia para el futuro desarrollo de las misiones espaciales en cuestión.

ACKNOWLEDGEMENTS

First of all, I am thankful to my supervisors for giving me the opportunity to carry out this thesis.

Moreover, I am very grateful to the hosts of my secondment in Milan, Juan Luis and Camilla. It was an honor to collaborate with them. And, of course, thanks to Hodei for making this secondment possible.

I would also like to acknowledge my colleagues for the breakfasts, lunches and occasional beers shared. Their company has been invaluable and one of the highlights of my thesis.

Finally, and most importantly, I am in deepest gratitude to my family and friends. Without their relentless support, this thesis would not have come to fruition.

PUBLICATIONS

The research carried out throughout the development of the present thesis has led to the publication or presentation of the following contributions.

JOURNAL ARTICLES

1. A. Barea, H. Urrutxua, and L. Cadarso. "Large-scale object selection and trajectory planning for multi-target space debris removal missions." In: *Acta Astronautica* (2020).
2. A. Barea, H. Urrutxua, and L. Cadarso. "Relative Inclination Strategy for J2-Perturbed Low-Thrust Transfers Between Circular Orbits." In: *Journal of Guidance, Control, and Dynamics* (2022).
3. A. Barea, H. Urrutxua, and L. Cadarso. "Dual-Based Method for Global Optimization of Impulsive Orbital Maneuvers." In: *The Journal of the Astronautical Sciences* (2022).
4. A. Barea, J. L. Gonzalo, C. Colombo, and H. Urrutxua. "A Constraint Programming framework for preliminary mission analysis: Applications for constellation-servicing active debris removal." *Submitted for publication*.

CONFERENCE PROCEEDINGS

1. A. Barea, H. Urrutxua, and L. Cadarso. "Optimal large-scale object selection and trajectory planning for active space debris removal missions." In: *8th European Conference for Aeronautics and Space Sciences* (2019).
2. A. Barea, H. Urrutxua, and L. Cadarso. "A branch-and-bound method for the global optimization of active debris removal missions." In: *3rd IAA Conference on Space Situational Awareness* (2022).
3. A. Barea, H. Urrutxua, and L. Cadarso. "Preliminary Analysis of a Fuel Station Strategy for Active Debris Removal Missions in Low Earth Orbit." In: *2022 AAS/AIAA Astrodynamics Specialist Conference* (2022).

1. A. Barea. "Optimización de trayectorias para eliminación activa de basura espacial." In: *I Congreso de la Escuela Internacional de Doctorado de la Universidad Rey Juan Carlos* (2018).

2. A. Barea, H. Urrutxua, and L. Cadarso. "Optimal large-scale object selection and trajectory planning for active space debris removal missions." In: *4th International Workshop on Key Topics in Orbit Propagation Applied to Space Situational Awareness* (2019).

3. A. Barea. "Diseño de trayectorias de bajo empuje para eliminación de basura espacial." In: *II Congreso de la Escuela Internacional de Doctorado de la Universidad Rey Juan Carlos* (2019).

4. A. Barea. "Diseño preliminar de misiones de eliminación activa de basura espacial." In: *III Congreso de Ingeniería Espacial del Instituto de la Ingeniería de España* (2020).

5. A. Barea, H. Urrutxua, and P. Solano-Lopez. "A simple indirect optimization method for J2-perturbed very-low-thrust transfers between circular orbits." In: *31st Workshop on JAXA Astrodynamics and Flight Mechanics* (2021).

6. A. Barea. "Misiones de eliminación de basura espacial: Mantenimiento del ambiente espacial y megaconstelaciones." In: *SpaceCon 21: Congreso Universitario del Espacio* (2021).

7. A. Barea, J.L. Gonzalo, C. Colombo, H. Urrutxua. "Preliminary analysis of an active debris removal mission for large constellations: A Constraint Programming methodology." In: *5th International Workshop on Key Topics in Orbit Propagation Applied to Space Situational Awareness* (2022).

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Part I

INTRODUCTION

It is a laborious madness and an impoverishing one,
the madness of composing vast books;
setting out in five hundred pages an idea that can be
perfectly related orally in five minutes.
The better way to go about it is to pretend that those
books already exist, and offer a summary,
a commentary on them...
A more reasonable, more inept, and more lazy man,
I have chosen to write notes on imaginary books.

— Jorge Luis Borges, The Garden of Forking Paths, 1962

1 INTRODUCTION

1.1 THE SPACE DEBRIS PREDICAMENT

Since the commissioning of the Sputnik 1 satellite in 1957, thousands of objects have been launched into Earth orbit. Most of those objects are concentrated in certain privileged orbital regions with properties that favour the scientific and commercial exploitation of the space operations. A high density of objects in these regions can result in collisions that generate new objects, thus increasing the possibility of subsequent collisions, and potentially leading to a cascade effect that can severely impact future space operations [65].

Other alarming phenomena that remarkably endanger the sustainability of the future space operations are the break-ups of spaceborne objects. Such explosions are mainly produced by the chemical instability of batteries and left-over fuel, but several cases of deliberate military-related cases have also been carried out [60]. In particular, one of the most concerning break-up incidents was the spontaneous explosion of an Ariane 1 upper stage in 1986, which generated the largest debris cloud to that date and led to the eventual creation of the Inter-Agency Space Debris Coordination Committee (IADC) in 1993 [61].

The purpose of the IADC is to coordinate the activities regarding space debris as well as to cooperate with member space agencies to identify possible debris mitigation strategies. Specifically, the IADC published the first version of its debris mitigation guidelines in 2002 [37]. Those guidelines include recommendations to limit the debris released during normal operations, minimize the potential for on-orbit break-ups, to carry out the post mission disposal and to prevent on-orbit collisions.

However, further research has concluded that those mitigation strategies are not enough to achieve the stabilisation of the number of resident space objects in the most populated zones. Hence, those mitigation strategies have to be complemented with the active removal of a number of high-criticality pieces of debris [71, 75]. This, along with the occurrence of two catastrophic events in the late 2000s, has motivated the research on the feasibility of carrying out active debris removal missions.

The first of such catastrophic events was the Chinese test of an anti-satellite weapon against the FengYun 1C weather satellite in 2007. As a result of this test, more than 2000 trackable fragments were generated. In turn, the NASA Orbital Debris Program Office estimated that

it produced over 35000 pieces of debris with a diameter larger than 1 centimeter [63]. The second of such events is the collision between the Iridium 33 and Cosmos 2251 satellites in 2009. This incident constitutes the first known collision of two satellites in orbit, where more than 1000 trackable fragments were produced [64].

Currently, the feasibility of diverse technologies that will allow to remove single pieces of space debris is being evaluated. In particular, the RemoveDebris mission, launched the 2nd of April 2018, performed in-orbit demonstrations of net and harpoon capture, as well as vision-based navigation [1, 46]. Moreover, the ELSA-d mission, launched the 22nd of March 2021 and currently operating, is performing demonstrations of capture with a magnetic docking mechanism and rendezvous and proximity operation technologies [20, 21]. Finally, the ClearSpace-1 mission is expected to demonstrate a series of debris removal technologies by means of removing a VEGA Secondary Payload Adapter upper part, with a total mass greater than 100 kg, by no later than the end of 2025 [19].

1.2 THESIS OBJECTIVES

As previously stated, current missions are focusing on demonstrating the feasibility of removing single pieces of space debris. However, it is expected that future missions will target several objects with a single servicing satellite. Thus, the preliminary design of multi-target debris removal missions has become the object of extensive research [28, 30, 31, 38, 40, 43, 44, 58, 73, 74, 76, 77, 83, 85, 95, 99, 105–108, 115]. In particular, the preliminary design of these missions involves the selection of the objects to be removed, their associated removal sequence, as well as the necessary maneuvers that the servicing satellite has to perform to carry out the mission.

Interestingly, the resolution of problem instances that consider a large candidate pool of objects has not been thoroughly explored in the aforementioned literature. This provides a fruitful line of research, as the development of methods capable of dealing with a pool of objects that is large enough to be representative of the distribution of the most hazardous objects in the region of interest, would provide a more realistic view of the actual deorbiting capabilities of multi-target missions.

Specifically, the efficient resolution of such large-scale instances poses three particular challenges. The most evident of such difficulties is the combinatorial complexity resulting from the size of the candidate object pool. Hence, a thorough exploration of all the possible object combinations requires the development of a methodology to extensively prune the combinatorial search space.

Moreover, the determination of the feasibility of each of the explored mission sequences involves of the the optimization of the ma-

neuvers to transfer between each of the concerning pairs of objects. In turn, such optimization requires the repeated evaluation of the orbital mechanics that governs the motion of each of the involved objects, thus generating a considerable computational burden. Therefore, it is of capital importance to devise maneuver optimization algorithms that achieve and advantageous trade-off between the fidelity of the orbital dynamics, the optimality of the transfers and the computational efficiency.

Finally, it has to be noted that the object selection and the maneuver optimization involve decisions of radically dissimilar nature. In particular, the object selection involves the determination of a set of discrete design variables. Conversely, the maneuver optimization entails the determination of a set of continuous design variables subject to the nonlinear and nonconvex model that describes the orbital dynamics. It turns out that the optimization of nonconvex models with discrete variables is a particularly difficult endeavor. Hence, the interaction between the object selection and the maneuver optimization is a challenge that the methods presented in this thesis have to overcome.

All in all, the methodologies developed in this thesis are able to successfully address the aforementioned challenges. Thus providing valuable mission design tools, not only for active debris removal missions, but also for other applications that entail decisions with characteristics similar to the ones considered in this thesis.

1.3 THESIS OUTLINE

The structure of this thesis comprises four different parts. Part i contains the present chapter and provides a general introduction to the content covered in this thesis.

Part ii introduces computationally efficient methods to optimize orbital transfers. Each of the chapters contained in this part correspond to a particular thesis publication:

- Chapter 2 (Ref. [10]): Optimization problems involving multiple impulsive maneuvers are, in general, nonlinear and nonconvex. This implies that their resolution is prone to local optimality and convergence issues. This work proposes an optimization method to exploit a specific structure of single-constraint nonlinear programming problems. The proposed algorithm is able to transform an optimization problem with an arbitrary number of variables into a root-finding problem of a univariate algebraic equation. Moreover, it can readily overcome the aforementioned local optimality and convergence issues. This methodology has been applied to three practical application examples. The first application involves the inclination optimization for change of plane maneuvers using drifting orbits with a relative nodal pre-

cession. The second application performs the semi-major axis optimization of phasing orbits, using a two-stage approach to solve it; specifically, the dual-based method yields a solution with phasing orbits that perform a fractional number of revolutions, which is then corrected to provide the appropriate integrality condition. The third application carries out the optimization of multi-impulse Hohmann-like transfers with an inclination change, relying on a conservation law that allows to compute a multi-impulse transfer from the solution of a two-impulse transfer. Finally, two highly relevant mission scenarios are described and numerically solved: the first scenario considers a geostationary transfer orbit with a phasing optimization to locate a satellite into a prescribed slot in geostationary orbit; the second scenario considers a multi-target rendezvous of a servicing spacecraft to visit several satellites of a constellation for debris removal or refueling operations.

- Chapter 3 (Ref. [12]): A novel J_2-perturbed continuous-thrust transfer between circular orbits is proposed. Specifically, this transfer considers tangential and out-of-plane thrust, with the thrust yaw angle being the sole control variable. Moreover, the dynamics of the transfer is described by two state variables, namely, the semimajor axis and the relative inclination with respect to the target orbit. The optimal solution of this problem involves finding a root of a function of the thrust yaw angle. Two heuristic control laws provide initial guesses for this root-finding problem. The convergence of this solution can be guaranteed with the use of bracketing root-finding methods, assuming that the function is continuous. Furthermore, a thrust-coast-thrust strategy has been formulated. Its first propulsive arc corresponds to an Edelbaum transfer, while the second one performs the aforementioned relative inclination change transfer. This results in a three-variable NLP problem. The performance of this strategy for time-constrainted Δv-optimal transfers has been compared with the solutions of other relevant methods available in the literature. It has been shown that the proposed three-arc strategy provides an advantageous trade-off between solution quality and computational complexity, thus being suitable for the formulation of large-scale combinatorial problems. Finally, it has been shown that the use of the proposed heuristic control laws in the second propulsive arc results in near-optimal solutions, while further accelerating the resolution process.

Part iii presents different frameworks for the preliminary analysis of active debris removal missions, with a special focus on the the efficient treatment of the combinatorial complexity and the interaction of the maneuver optimization with the combinatorial decisions. Each

of the chapters contained in this part correspond to a particular thesis contribution:

- Chapter 4 (Ref. [8]): Upcoming active space debris removal missions will most likely attempt to remove several objects per mission. The design of such missions involves the selection of the objects to be removed, as well as the optimisation of the visit sequence and the orbital transfers interconnecting them. In this chapter a branch-and-bound-based algorithm is presented for the preliminary design of multi-target space debris removal missions. The proposed algorithm comprises two different levels. The upper level, modelled as an Integer Linear Programming problem, deals with the combinatorial complexity of the problem. The lower level, modelled as a Mixed Integer Nonlinear Programming problem, encapsulates the orbital dynamics. Throughout the problem resolution, the upper level selects promising subsets of a pool of candidate objects of space debris, so that a removed threat value is maximised. Each of these subsets is passed through to the lower level, which ensures that there is a feasible trajectory that allows to rendezvous, in a specific sequence, with each and every object in the subset, while prescribed mission duration and Δv constraints are fulfilled. This framework is able to exploit the structure of the problem so that instances with large pools of candidate objects can be efficiently solved while achieving the certificates of optimality that branch-and-bound methods provide.

- Chapter 5 (Ref. [9]): The modelling of multi-target active debris removal missions requires the use of continuous as well as discrete decision variables. This, along with the nonlinearity of the orbital mechanics that governs the motion of the involved objects, results in highly nonlinear and nonconvex problems which are prone to the appearance of a great number of local optimal solutions. This chapter proposes an exact methodology, based on Mixed Integer Programming techniques, for the global optimization of multi-target active debris removal missions. Specifically, the proposed method integrates the combinatorial decisions (concerning the target and sequence selection) and the orbital mechanics (regarding the computation of the Δt and Δv spent during each of the transfers) into a single Mixed Integer Linear Programming (MILP) problem. It is necessary for the MILP formulation to comprise exclusively linear constraints, while the expressions that determine the consumed Δt and Δv are clearly nonlinear. Thus, a piecewise linear approximation of such expressions is considered. Branch-and-bound techniques are used to solve the resulting MILP problem. The linear relaxation of the model provides an efficient pruning of the search tree by means of the obtention of optimistic bounds for each

of the considered branches. This way, a global optimum of arbitrary precision, i.e., according to the number and position of the nodes used to approximate the Δt and Δv functions, is achieved.

- Chapter 6 (Ref. [11]): A two-stage methodology for the preliminary design of active debris removal missions comprising a servicing satellite and a fuel station is proposed. The upper stage explores promising values of the station location. The lower stage determines the objects to be removed, the removal sequence and the maneuvers of the servicing satellite so that the aggregated criticality of the selected objects is maximized. This is modelled as a Mixed Integer Linear Programming problem and solved using a branch-and-bound method. Both stages are iterated until the optimal station location is found. Two practical cases involving prominent debris clusters have been analyzed.

- Chapter 7 (Ref. [7]): Constraint Programming is a classical artificial intelligence paradigm characterised by its flexibility for the modelling of complex problems. In the field of space operations, this approach has been successfully used for mission planning and scheduling. This chapter proposes a framework that leverages the strengths of Constraint Programming for the preliminary analysis of space missions. Specifically, it uses constraint propagation and search techniques to thoroughly explore the configuration space of a mission in an efficient manner. Consequently, it is able to quantify the performance of precomputed mission choices with respect to the mission requirements, as well as generate new ones that optimise such performance. The proposed methodology has been particularised for two application cases involving active debris removal missions for large constellations in low Earth orbit, namely, a chaser case and a mothership case. The chaser case considers a servicing satellite that rendezvouses with the failed satellites of the constellation and directly transports them to a disposal orbit. The mothership case comprises a servicing satellite that installs deorbiting kits in each of the failed satellites, except for the one removed in the last place. This way, the servicing satellite will only transport this object, while the deorbiting kits will carry out the disposal of the rest of them. This methodology has been successfully used to evaluate a preliminary mission analysis of both application cases developed under ESA's Sunrise project.

Finally, Part iv contains Chapter 8 and exposes the global conclusions that can be drawn from the content included in the preceding Parts.

Part II

MANEUVER OPTIMIZATION

Go uses the most elemental materials and concepts; line and circle, wood and stone, black and white; combining them with simple rules to generate subtle strategies and complex tactics that stagger the imagination.

— Iwamoto Kaoru, professional Go player

2

DUAL-BASED METHOD FOR IMPULSIVE ORBITAL MANEUVERS

2.1 INTRODUCTION

In the literature there are numerous, well-proven algorithms to solve general Nonlinear Programming (NLP) problems [15]. However, specialized algorithms that exploit particular problem structures, albeit limited in scope, can provide a meaningful edge with respect to general purpose methods. Specifically, they may enable to solve the problems at hand more efficiently, or provide an insight on the mathematical properties of the optimal solution.

One of the most prominent mathematical tools to develop such specialized algorithms is the concept of duality [51]. It entails that, given a constrained mathematical programming problem (referred to as primal problem), there exists another optimization problem (referred to as dual problem) whose optimal solution is an optimistic bound of the optimal solution of the primal problem. Moreover, if the primal problem is convex, both problems achieve the same solution. The relation between such problems lies in the fact that the optimization variables of said dual problem are the Lagrange multipliers of the constraints of the primal problem, which are used to define the Lagrangian function of the latter and, consequently, appear in its optimality conditions. Hence, the mathematical properties derived from duality are particularly useful for the resolution of problems with a small number of constraints, as they generate dual problems with such number of optimization variables.

This work proposes an optimization method to exploit a specific structure of single-constraint NLP problems. The proposed algorithm is able to transform an optimization problem with an arbitrary number of variables into a root-finding problem of a univariate algebraic equation. The variable to be solved for is the dual variable (i.e., the Lagrange multiplier) of the single constraint of the problem, from which the optimal solution of the problem is retrieved. Consequently, this technique can be referred to as a dual-based method. This term has often been used in literature to refer to optimization algorithms that take advantage of the resolution of the dual problem; this led to a variety of such methods applied to classical optimization problems like facility location problems [42], network design problems [4], transportation problems [92] and assignment problems [54].

More recently, dual-based methods have proven useful for diverse applied optimization problems such as the reconstruction of sparse

signals [102], the operation of telecommunication networks [56] or the management of chemical reactors [79].

However, the proposed methodology is essentially distinct in nature from the aforementioned techniques, in the sense that it is not the dual problem what is being solved, but instead the primal problem, only that it is formulated in terms of the dual variable. This is particularly beneficial because, for general nonlinear problems, the optimal solution of the dual problem is not the primal solution, but an optimistic bound of it. Thus, the proposed method is able to obtain the optimal solution while taking advantage of the properties of duality, since it is the primal problem that is being solved all along.

Interestingly enough, many optimization problems in orbital mechanics, specifically those related to orbital transfers with impulsive maneuvers, can actually be reduced to a mathematical model that abides by the described structure. Therefore, optimization problems such as fuel-minimum Hohmann-like transfers combined with inclination changes and phasing maneuvers, or the optimization of time-minimal sequences of phasing maneuvers or change of plane maneuvers with a prescribed ΔV budget, are but a few example applications that can benefit from the proposed approach. Although it is acknowledged that these are classical problems for which solutions and algorithms already exist in the literature, the methodology presented in this manuscript still provides a novel, alternative approach that offers an insightful, complementary view to these problems, by proposing a methodology that can be easily and conveniently extended to other optimization problems that conform to the same mathematical structure of a single-constraint NLP problem.

After the derivation of the proposed mathematical methodology is detailed in Section 2.2, three practical application examples of this methodology are presented in Section 2.3, based on typical impulsive maneuver optimization problems, namely: 1) the inclination optimization for change of plane maneuvers using drifting orbits with a relative nodal precession; 2) the semi-major axis optimization of phasing orbits; and 3) the optimization of combined, multi-impulse Hohmann-like transfers with an inclination change. To illustrate the applicability of the proposed methodology to mission-related problems, the aforementioned application examples are combined and re-purposed to solve two relevant mission scenarios, which are described and numerically solved in Section 2.4: the first scenario considers a geostationary transfer orbit with a phasing optimization to locate a satellite into a prescribed slot in geostationary orbit; the second scenario considers a multi-target rendezvous of a servicing spacecraft to visit several satellites of a constellation. Finally, Section 2.5 summarizes the main conclusions of this work.

2.2 MATHEMATICAL MODEL AND SOLUTION APPROACH

The proposed approach is focused on processes that comprise different stages, which may be indexed by $k \in K$. Each of these stages has a dependence on a single decision variable, a_k, and the aim is to determine the values of the decision variables such that a performance index F is minimized while complying with a given cost G.

The aforementioned problem can be modelled as an optimization problem featuring the performance index, F, as objective function:

$$F = \min \sum_{k \in K} f_k(a_k), \tag{1}$$

where functions $f_k(a_k)$ denote the contribution of each of the stages to the performance index. The objective function of the problem is also subject to an equality constraint:

$$\sum_{k \in K} g_k(a_k) = G, \tag{2}$$

where $g_k(a_k)$ are the contributions of each of the stages to the cost G.

The Lagrangian function of this problem can be stated as

$$\mathcal{L}(a_1, a_2, \ldots, a_{|K|}, \lambda) = \sum_{k \in K} f_k(a_k) - \lambda \left(\sum_{k \in K} g_k(a_k) - G \right), \tag{3}$$

where λ is the dual variable of Eq. (2), and thus the optimality conditions of the problem are obtained by nullifying the gradient of the Lagrangian function, which yields the following system of scalar equations:

$$\frac{df_k}{da_k} - \lambda \frac{dg_k}{da_k} = 0, \quad \forall k \in K \tag{4a}$$

$$\sum_{k \in K} g_k(a_k) - G = 0. \tag{4b}$$

If functions g_k are invertible, then their inverse functions, $g_k^{-1}(a_k) = a_k(g_k)$, allow to transform $f_k(a_k)$ into $f_k(g_k)$, and consequently, Eq. (4a) becomes:

$$\frac{df_k}{dg_k} = \lambda, \quad \forall k \in K \tag{5}$$

Eq. (5) has a clear interpretation: the derivative of the performance index with respect to the cost is constant for every stage, and this constant turns out to be the dual variable, λ. Interestingly, this property can be exploited to obtain the solution of the proposed problem, basically by determining the value of the dual variable. To this end,

the cost function of each of the stages, $g_k(a_k)$, may be reformulated as a function of λ:

$$\text{inv}\left(\frac{df_k}{dg_k}(g_k)\right) = g_k\left(\frac{df_k}{dg_k}\right) = g_k(\lambda). \tag{6}$$

Replacing $g_k(a_k)$ by $g_k(\lambda)$ into the left-hand side of Eq. (2), the following univariate function is obtained:

$$\phi(\lambda) = \sum_{k\in K} g_k(\lambda) - G \tag{7}$$

Obviously, a root of Eq. (7) readily provides an optimal solution to the problem, as it simultaneously fulfills the optimality conditions given by Eqs. (4). Consequently, the single-variable equation $\phi(\lambda) = 0$ as defined by Eq. (7), becomes the new optimality condition, which can be numerically solved for the dual variable λ using a suitable root-finding algorithm. Finaly, $g_k(\lambda)$ can be introduced into the function $a_k(g_k)$ to recover the decision variables that describe the original optimization problem.

It is worth nothing that, despite the problem defined by Eqs. (1-2) being readily solved by general NLP techniques, the methodology proposed in this manuscript improves on two usual difficulties proper to the NLP approach: the convergence of the method, and the local optimality of non-convex problems. The dual-based method approach circumvents the convergence issues by relying on bracketing root-finding algorithms (e.g., Brent's method [24]), which guarantee the convergence for bounded and continuous functions $\phi(\lambda)$. Regarding the optimality of non-convex problems, the fact that multiple local optima exist implies that, in the dual-based approach, the inverse of any of the df_k/dg_k functions may not be unique; however, as opposed to NLP problems, the dual-based approach still provides an edge, since a careful study of these functions based on well-known functional analysis techniques provides means of removing this multiplicity issue, and selecting the correct inverse functions that do result in the global optimum. Therefore, the dual-based method does offer interesting advantages over a classical NLP approach.

2.3 PRACTICAL APPLICATIONS EXAMPLES

This section details the application of the proposed dual-based optimization method to three types of impulsive orbital transfers, which can be mathematically described as a single-constraint optimization problem, thus allowing the application of the proposed methodology. Although these problems are simple and of a mainly academic interest, they illustrate the use of the dual-based methodology, which can be extended to optimization problems of higher complexity and relevance, as will be shown later in Section 2.4.

2.3.1 *Inclination optimization for minimum-time orbital drift*

In this first application, a spacecraft is intended to visit a series of k orbits, identical in shape to the departure orbit, but shifted in their right ascension of the ascending node, such that the difference between one orbit (k) and its preceding one ($k-1$) is represented by $\Delta\Omega_k$. The nodal precession produced by the Earth's J_2 perturbation is exploited to accomplish these shifts, so the required change of relative precession is achieved by performing an impulsive maneuver that modifies the inclination of the spacecraft orbit, thus varying its nodal precession rate and allowing it to advance until the right ascension of the next target orbit is matched. This modified orbit is referred as a *drifting* orbit. The spacecraft has a predefined ΔV_{tot} budget, which has to be allocated to the different impulses such that the mission time is minimized.

The ΔV required to perform an inclination change of Δi is given by

$$\Delta V = 2\,V \sin\left(\frac{\Delta i}{2}\right), \tag{8}$$

where V stands for the orbital velocity at the moment of performing the impulse. The time, Δt, that it takes the spacecraft to reach a target orbit with a right ascension $\Delta\Omega$ away is given by

$$\Delta t = \frac{\delta\,\Delta\Omega}{\dot{\Omega}_{tgt} - \dot{\Omega}_{drift}}, \tag{9}$$

where $\dot{\Omega}$ stands for the nodal precession, and the subindices 'tgt' and 'drift' label the target orbit and the drifting orbit, respectively. The parameter δ has been introduced so that the quantities ΔV, Δt, Δi and $\Delta\Omega$ are always positive; hence, this parameter takes care of the appropriate sign depending on whether the drifting orbit is leading ($\delta = 1$) or trailing ($\delta = -1$) the target orbit in $\Delta\Omega$.

The nodal precessions of the target orbit and the drifting orbit, respectively are defined as:

$$\dot{\Omega}_{tgt} = -\frac{3}{2}\frac{n R_\oplus^2 J_2}{p^2}\cos(i), \tag{10}$$

$$\dot{\Omega}_{drift} = -\frac{3}{2}\frac{n R_\oplus^2 J_2}{p^2}\cos(i - \delta\,\Delta i), \tag{11}$$

where i is the inclination of the target orbit, n stands for the mean orbital motion, p is the semilatus rectum, $R_\oplus$ is the reference radius of the Earth, and J_2 is the coefficient associated to the degree 2 and order 0 spherical harmonic of the Earth's gravity field.

This optimization problem can be formulated with the dual-based methodology by identifying the Δt required for each maneuver with the functions $f_k(a_k)$, i.e., the contributions to the objective function

F as defined in Eq. (1), which here represents the total mission time it takes to visit all target orbits; the role of the decision variables, a_k, is here played by the variable Δi, that defines each of the inclination change maneuvers. Likewise, the cost for each of these maneuvers, represented by their ΔV, can be matched to the functions $g_k(a_k)$ that comprise the equality constraint of Eq. (2), so that the aggregated ΔV for all the considered maneuvers is equal to the ΔV budget, i.e., $G = \Delta V_{tot}$.

By differentiating Eqs. (8-11) and using the chain rule, the dual variable λ can be readily obtained:

$$\lambda = \frac{d\Delta t}{d\Delta V} = \frac{d\Delta t}{d\dot{\Omega}_{drift}} \frac{d\dot{\Omega}_{drift}}{d\Delta i} \frac{d\Delta i}{d\Delta V}, \tag{12}$$

where each of the factors is given by the following expressions:

$$\frac{d\Delta t}{d\dot{\Omega}_{drift}} = \frac{\delta\,\Delta\Omega}{\left(\dot{\Omega}_{tgt} - \dot{\Omega}_{drift}\right)^2} \tag{13}$$

$$\frac{d\dot{\Omega}_{drift}}{d\Delta i} = -\frac{3}{2} \frac{n R_\oplus^2 J_2}{p^2}\, \delta \sin(i - \delta\,\Delta i) \tag{14}$$

$$\frac{d\Delta i}{d\Delta V} = \frac{1}{V \cos\left(\frac{\Delta i}{2}\right)}. \tag{15}$$

Thus, the dual variable λ is naturally expressed as an implicit function of the variable Δi and, ultimately, of the variable ΔV through Eq. (8). Therefore, particularizing Eq. (7) to this problem yields:

$$\phi(\lambda) = 2 \sum_{k \in K} \Delta V_k(\lambda) - \Delta V_{tot} \tag{16}$$

where $\Delta V_k(\lambda)$ is the inverse function of Eq. (12) for each of the k target orbits. These functions are multiplied by a factor 2 because the inclination change is performed twice, i.e., once to initiate the relative nodal precession, and for a second time when the target orbit is reached in order to match its inclination and nullify the relative nodal precession rate. In brief, solving the algebraic equation of Eq. (16) with a suitable root-finding technique readily provides the optimal solution of the problem; once its root is computed, evaluating Eqs. (8) and (12) allows to recover the optimal inclination changes (i.e., the decision variables), Δi, along with their associated ΔV for visiting each of the target orbits in the minimum time with the available propellant.

A simple numerical example is subsequently discussed so as to illustrate the aforementioned resolution process, as well as to obtain additional insight about the qualitative behavior of $d\Delta t/d\Delta V$ as computed by Eq. (12). Specifically, given a circular orbit around the Earth with an altitude of 550 km and an inclination of 53 deg, the problem at hand involves the optimal distribution of a ΔV budget of 50 $m\,s^{-1}$ between two consecutive drifting maneuvers with associated $\Delta\Omega$ values of 5 and 10 deg, respectively, such that the aggregated transfer

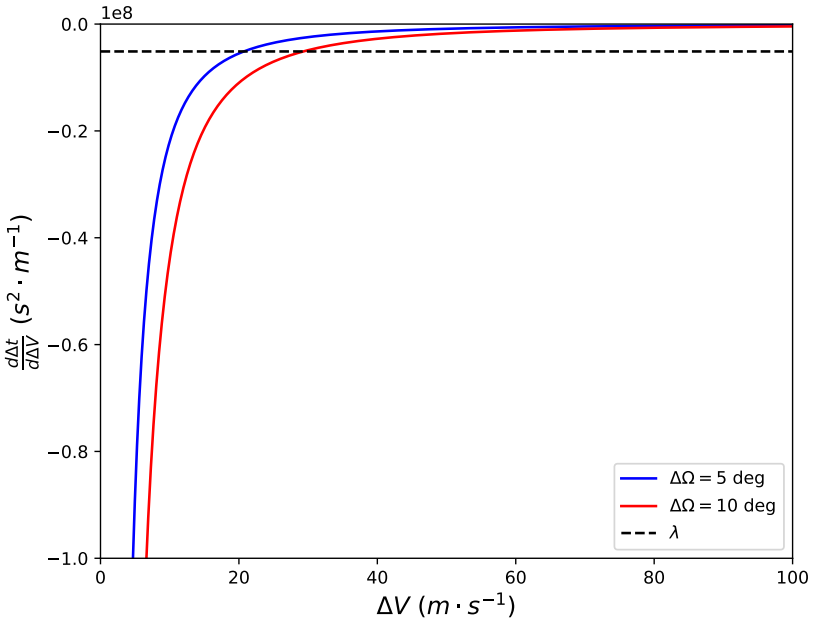


Figure 1: Solution of the orbital drift numerical example

time is minimized. Moreover, the initial orbit is trailing both target orbits in right ascension of the ascending node and, likewise, the first target orbit is trailing the second one. Hence, δ is given a value of -1 for both transfers.

Fig. 1 shows, for each of the transfers, $d\Delta t/d\Delta V$ as a function of the ΔV spent to generate the relative nodal precession as well as to subsequently match the inclination of the target orbit (i.e., $2\Delta V_k$, as shown in Eq. (16)). Furthermore, solving Eq. (16) results in $\lambda = -5.124978 \cdot 10^6$ $s^2\,m^{-1}$, which is represented by the dashed black line. It can be seen that the intersection of this line with the aforementioned functions provides the optimal ΔV allocation for their corresponding transfers. Further details of the solution are provided in Table 1.

Table 1: Numerical example for the minimum-time orbital drift application.

$\Delta\Omega$ [deg]	ΔV [$m\,s^{-1}$]	Δi [deg]	Δt [s]
5	20.71	$7.818 \cdot 10^{-2}$	$5.3043635 \cdot 10^7$
10	29.29	$1.106 \cdot 10^{-1}$	$7.4999114 \cdot 10^7$

Finally, the qualitative behavior of the functions showed in Fig. 1 can be compared with the individual tendencies of the derivatives described by Eqs. (13-15) to obtain a better insight about the structure of $d\Delta t/d\Delta V$ (as defined in Eq. (12)). At first sight, it seems that the

functions in Fig. 1 asymptotically approach minus infinity and zero when ΔV tends to zero and infinity, respectively.

On the one hand, if ΔV tends to zero, so does Δi. In that case, Eqs. (10-11) determine that $\dot{\Omega}_{drift}$ and $\dot{\Omega}_{tgt}$ are identical. Thus, the derivative in Eq. (13) tends to minus infinity, while the derivatives in Eqs. (14-15) tend to positive finite numbers. Therefore, $d\Delta t/d\Delta V$ has a vertical asymptote when ΔV tends to 0.

On the other hand, according to Eq. (8), ΔV has a finite maximum value so it is not possible for $d\Delta t/d\Delta V$ to have the hypothesized horizontal asymptote. The maximum ΔV is achieved when $\Delta i = \pi$. In that case, the derivative in Eq. (15) tends to infinity while the derivatives in Eqs. (13-14) tend to negative finite numbers. Therefore, $d\Delta t/d\Delta V$ has a vertical asymptote when ΔV tends to $4V$ (i.e., twice the maximum ΔV of Eq. (8) to represent the maximum value of $2\Delta V_k$). However, the positive values of $d\Delta t/d\Delta V$ are not relevant for this analysis, as they entail that spending additional ΔV will increase the mission time. Therefore, the maximum relevant ΔV for this analysis is the one that nullifies $d\Delta t/d\Delta V$. In particular, this ΔV magnitude results from imposing that $\Delta i = \pi - i$, thus nullifying the derivative in Eq. (14) while the derivatives in Eqs. (13,15) are finite numbers.

2.3.2 *Semi-major axis optimization for minimum-time orbit phasing*

In this application, a spacecraft initially located on a circular orbit has to perform a rendezvous with k target spacecraft located on the same orbit, but shifted in mean anomaly. The shift between one object (k) and its preceding one ($k-1$) is represented by ΔM_k. A relative mean motion between the spacecraft and the targets can be triggered to perform the needed orbital repositioning maneuvers. Each of these phasing maneuvers is achieved with an impulsive burn that modifies the semi-major axis of the spacecraft orbit, which results in an elliptic orbit, tangent to the original one, referred to as a *phasing* orbit, where the spacecraft will coast until the phase difference with the target spacecraft is nullified; at that moment, another impulse is applied to recover the initial semi-major axis, thus returning the departure orbit in phase with the target. The spacecraft has a predefined ΔV_{tot} budget, which has to be allocated to the different impulses so that the mission time is minimized.

The ΔV spent to move from the target's orbit into the the phasing orbit, and back to the target's orbit (once the phasing is complete), is given by:

$$\Delta V = 2\delta \left(\sqrt{\frac{2\mu}{a_{tgt}} - \frac{\mu}{a_{pha}}} - \sqrt{\frac{\mu}{a_{tgt}}} \right), \tag{17}$$

where a_{tgt} and a_{pha} are the semi-major axes of the target and phasing orbits, respectively, and μ stands for the gravitational parameter;

the parameter δ, as in Section 2.3.1, is used to make all increments positive, and its value depends on whether the spacecraft is leading ($\delta = 1$) or trailing ($\delta = -1$) the considered target spacecraft in mean anomaly ΔM. The time it takes the spacecraft to intercept a given target, Δt, is given by:

$$\Delta t = \frac{\delta \, \Delta M}{n_{tgt} - n_{pha}}, \tag{18}$$

where n_{tgt} and n_{pha} are the mean orbital motions of the target and phasing orbits, respectively.

Again, this optimization problem can be formulated with the dual-based methodology by identifying the Δt of each phasing maneuver with the functions $f_k(a_k)$, where decisions variables are now described by the semi-major axis of each of the phasing orbits, namely a_{pha}. Likewise, the ΔV of each of these phasing maneuvers can be matched to the functions $g_k(a_k)$ in the equality constraint of Eq. (2), so that the aggregated ΔV for all the considered maneuvers is equal to the ΔV budget, i.e., $G = \Delta V_{tot}$.

Proceeding as in Section 2.3.1, the derivative chain rule provides the dual variable as follows:

$$\lambda = \frac{d\Delta t}{d\Delta V} = \frac{d\Delta t}{dn_{pha}} \frac{dn_{pha}}{da_{pha}} \frac{da_{pha}}{d\Delta V} \tag{19}$$

where each of the factors are given by:

$$\frac{d\Delta t}{dn_{pha}} = \frac{\delta \Delta M}{\left(n_{tgt} - n_{pha}\right)^2} \tag{20}$$

$$\frac{dn_{pha}}{da_{pha}} = -\frac{3 n_{pha}}{2 a_{pha}} \tag{21}$$

$$\frac{da_{pha}}{d\Delta V} = \frac{8 a_{tgt}^2 \, \mu \left(2\delta \sqrt{\frac{\mu}{a_{tgt}}} + \Delta V\right)}{\left(4 a_{tgt} \, \Delta V \, \delta \sqrt{\frac{\mu}{a_{tgt}}} + a_{tgt} \, \Delta V^2 - 4\mu\right)^2}. \tag{22}$$

Thus, the dual variable λ is naturally expressed as an implicit function of the semi-major axis of the phasing orbit, a_{pha}, which is ultimately a function of the variable ΔV through Eq. (17). Therefore, particularizing Eq. (7) to this problem yields

$$\phi(\lambda) = \sum_{k \in K} \Delta V_k(\lambda) - \Delta V_{tot} \tag{23}$$

where $\Delta V_k(\lambda)$ is the inverse function of Eq. (19) for each of the k target spacecraft. Solving for the root of Eq. (23) readily provides the optimal solution of the problem. However, this solution neglects the fact that the spacecraft can only complete an integer number of revolutions in the phasing orbit before it can return to the target orbit.

Hence, this solution has to be corrected to take into account this additional constraint.

The number of fractional revolutions, $C \in \mathbb{R}$, that the spacecraft spends in the phasing orbit, is computed as:

$$C = \frac{\Delta t}{T_{pha}} . \tag{24}$$

Since the number of revolutions must be an integer number, assigning a semi-major axis corresponding to the ceiling of C, for each of the k phasing orbits, would readily result in a feasible solution. However, the new phasing orbits would require less ΔV than the original ones, resulting in an unused ΔV surplus. This spare ΔV could be reassigned among the phasing orbits such that a feasible solution could be obtained, which further minimizes the mission duration.

In particular, the problem at hand is to decide, for each phasing orbit, whether the satellite spends a number of revolutions $\text{ceil}(C)$ on the phasing orbit. This can be modelled with the following Integer Linear Programming problem:

$$\min \sum_{k \in K} \Delta \tilde{t}_k \, X_k \tag{25a}$$

$$\sum_{k \in K} \Delta \tilde{V}_k \, X_k \leqslant \Delta V_{sur} \tag{25b}$$

where X_k is a binary variable that is either 0 or 1, depending on whether the number of revolutions performed at the orbit k corresponds to the ceiling or the floor of C, respectively; $\Delta \tilde{t}_k$ and $\Delta \tilde{V}_k$ are the differences of Δt and ΔV, respectively, resulting from choosing $X_k = 1$ instead of $X_k = 0$; and ΔV_{sur} is the ΔV surplus obtained when selecting $X_k = 0$ for every orbit. This kind of problems is known as a Knapsack problem [78].

Nevertheless, the particular properties of $\Delta \tilde{t}_k$ can be exploited to further simplify this correction problem. Substituting Δt from Eq. (18) into Eq. (24) and isolating T_{pha} results in the following relation:

$$T_{pha} = T_{tgt} \left(1 + \frac{\delta \, \Delta M}{2\pi \, C} \right) . \tag{26}$$

According to Eq. (24), multiplying both sides of Eq. (26) by C allows to obtain Δt as a function of C, i.e.:

$$\Delta t(C) = T_{tgt} \left(C + \frac{\delta \, \Delta M}{2\pi} \right) . \tag{27}$$

Thus, $\Delta \tilde{t}_k$ can be obtained by subtracting $\Delta t(C)$ for two values of C that differ in one revolution, which yields:

$$\Delta \tilde{t}_k = \Delta t(C) - \Delta t(C+1) = -T_{tgt} \quad \forall k \in K . \tag{28}$$

Interestingly, the latter equations reveals that $\Delta\tilde{t}_k$ does not depend on the parameters of the phasing orbit, nor the number of revolutions necessary to perform the rendezvous. Thus, $\Delta\tilde{t}_k$ is constant, i.e., it is the same for the k phasing orbits, and therefore it is not necessary to solve the Knapsack problem of Eqs. (25) to obtain the corrected solution. In brief, this solution can be directly obtained by ordering the phasing orbits in increasing value of $\Delta\tilde{V}_k$ and making $X_k = 1$ in that order, for as long as Eq. (25b) remains feasible.

A simple numerical example is subsequently discussed so as to illustrate the resolution process of this application, as well as to obtain additional insight about the qualitative behavior of $d\Delta t/d\Delta V$ as computed by Eq. (19). Specifically, given a circular orbit around the Earth with an altitude of 550 km, the problem at hand involves the optimal distribution of a ΔV budget of 5 $m\,s^{-1}$ to perform two consecutive phasing maneuvers with associated ΔM values of 16.36 and 32.72 deg, respectively, such that the aggregated transfer time is minimized. Moreover, the initial position of the spacecraft is trailing both targets and, likewise, the first target is trailing the second one. Hence, δ is given a value of -1 for both phasing maneuvers.

Fig. 2 shows, for each of the phasing maneuvers, $d\Delta t/d\Delta V$ as a function of the ΔV spent to perform such maneuvers, as defined in Eq. (17). Furthermore, solving Eq. (23) results in $\lambda = -3.072195 \cdot 10^5$ $s^2\,m^{-1}$, which is represented by the dashed black line. It can be seen that the intersection of this line with the aforementioned functions provides the optimal ΔV allocation for their corresponding transfers. Further details of the solution are provided in Table 2. It can be seen that the computed phasing maneuvers require a fractional number of revolutions, which is not feasible. Therefore, the solution has been adjusted using the previously described correction algorithm, resulting in the corrections displayed in Table 3. Hence, the feasible solution can be obtained by means of the addition of these corrections to the infeasible one. It has to be noted that the computed ΔV corrections are very small, which indicates that the infeasible solution represents a good estimation of the optimal feasible solution.

Table 2: Numerical example for the minimum-time phasing application.

ΔM [deg]	ΔV [$m\,s^{-1}$]	a [m]	Δt [s]	Revolutions
16.36	2.07	$6.919122 \cdot 10^6$	$6.36269 \cdot 10^5$	111.08
32.72	2.93	$6.918340 \cdot 10^6$	$8.99822 \cdot 10^5$	157.12

Finally, the qualitative behavior of the functions showed in Fig. 2 can be compared with the individual tendencies of the derivatives described by Eqs. (20-22) to obtain a better insight about the structure of $d\Delta t/d\Delta V$ (as defined in Eq. (19)). At first sight, it seems that the

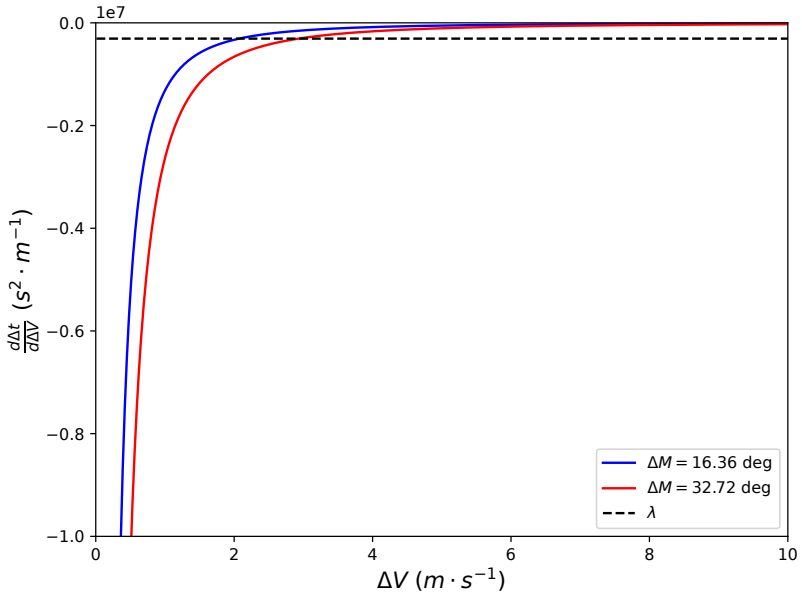


Figure 2: Solution of the phasing numerical example

Table 3: Revolution correction of the minimum-time phasing application.

ΔM [deg]	ΔV Corr. [m s^{-1}]	a Corr. [m]	Δt Corr. [s]	Revolutions
16.36	$1.576 \cdot 10^{-3}$	$-1.436 \cdot 10^{0}$	$-4.8384 \cdot 10^{2}$	111
32.72	$-1.625 \cdot 10^{-2}$	$1.480 \cdot 10^{1}$	$5.0196 \cdot 10^{3}$	158

functions in Fig. 2 asymptotically approach minus infinity and zero when ΔV tends to zero and infinity, respectively.

On the one hand, if ΔV tends to zero, the target and phasing orbits are identical and so are their respective n_{tgt} and n_{pha}. In that case, the derivative in Eq. (20) tends to minus infinity, while the derivatives in Eqs. (21-22) tend to negative finite numbers. Therefore, $d\Delta t/d\Delta V$ has a vertical asymptote when ΔV tends to 0.

On the other hand, according to Eq. (17), ΔV has a finite maximum value of $2\sqrt{\mu/a_{tgt}}$ when $\delta = -1$. Thus, it is not possible for $d\Delta t/d\Delta V$ to have the hypothesized horizontal asymptote. If that ΔV magnitude is applied, the derivative in Eq. (22) vanishes, while the derivatives in Eqs. (20-21) are finite numbers. Consequently, the value of $d\Delta t/d\Delta V$ will also vanish in that particular instance. However, it has to be noted that the application of that ΔV magnitude results in an orbit with a null radius of perigee, hence being a degenerate case and not being relevant for this analysis. Specifically, the maximum relevant ΔV should be chosen such that it results in a phasing orbit with an operationally feasible radius of perigee.

2.3.3 *Optimization of multi-impulse, combined maneuvers*

It is well known that when performing a two-impulse, Hohmann-like transfer between two circular, non-coplanar orbits, the minimum ΔV is achieved when optimally distributing the relative inclination change between the two performed impulses [34, 100]. The technique described in Section 2.2 is able to solve this problem not only for the usual two-impulse transfer, but also for for the case in which each of these two impulses were divided into a set of *subimpulses*; these *subimpulses* achieve orbital velocities with values ranging between the circular and elliptic orbital velocities of the impulse they divide. This capability of the dual-based method provides a good opportunity to revisit this classical problem and analyze the relation between the two-impulse and multi-impulse transfers.

First, the solution of a problem that considers and arbitrary number of impulses is detailed. The ΔV spent at each of these impulses can be computed as a function of the Δi change they accomplish:

$$\Delta V(\Delta i) = \sqrt{V_1^2 + V_2^2 - 2V_1V_2\cos(\Delta i)}\,, \tag{29}$$

where V_1 and V_2, i.e., the orbital velocities before and after an impulse, are interchangeable; for the sake of notation, they are assigned such that $V_1 < V_2$. This optimization problem can be formulated with the dual-based methodology by identifying the ΔV at each impulse with the functions f_k, i.e., the contributions to the objective function F as defined in Eq. (1), which here represents the sum of the ΔV for all the performed impulses, namely ΔV_{tot}. Likewise, the change of inclination achieved by each impulse, Δi, can be matched to the functions g_k that comprise the equality constraint of Eq. (2), such that the sum of all considered impulses delivers the desired total change of inclination $G = \Delta i_{tot}$.

The dual-based method involves the derivative df_k/dg_k, which defines the dual variable λ following Eq. (5). This requires the derivation of Eq. (29) with respect to Δi, which yields:

$$\frac{d\Delta V}{d\Delta i}(\Delta i) = \frac{V_1V_2\sin(\Delta i)}{\sqrt{V_1^2 + V_2^2 - 2V_1V_2\cos(\Delta i)}} = \frac{V_1V_2\sin(\Delta i)}{\Delta V(\Delta i)}. \tag{30}$$

The latter equation vanishes for $\Delta i = \{0, \pi\}$ and has a maximum for $\Delta i = \arccos(V_1/V_2)$, which yields a value of $d\Delta V/d\Delta i = V_1$. This implies that the inverse of this function is not unique; as a matter of fact, the inverse has two possible branches. Defining the parameter $\rho = \{1, -1\}$ to identify each of these two branches, the inverse function can be written as follows:

$$\Delta i\left(\frac{d\Delta V}{d\Delta i}\right) = \Delta i(\lambda;\rho) = \arccos\left(\frac{\lambda^2 + \rho\sqrt{(\lambda^2 - V_1^2)(\lambda^2 - V_2^2)}}{V_1V_2}\right),$$

(31)

where $\lambda \in [0, V_1]$ is the dual variable and $\rho = 1$ corresponds to the increasing interval of $d\Delta V/d\Delta i$, namely $\Delta i \in [0, \arccos(V_1/V_2)]$, whereas $\rho = -1$ represents the decreasing interval after the maximum, i.e., $\Delta i \in [\arccos(V_1/V_2), \pi]$.

As a result, from the application of the dual-based method to this particular problem, Eq. (7) provides the following algebraic equations:

$$\phi(\lambda) = \sum_{k \in K} \Delta i_k(\lambda; \rho_k^*) - \Delta i_{tot}, \tag{32}$$

where the subscript k indexes the different impulses, and ρ_k^* stands for the right value of the parameter ρ that, for each impulse k, yields the global optimum of the problem. Thus, the values of ρ_k^* have to be determined for each impulse.

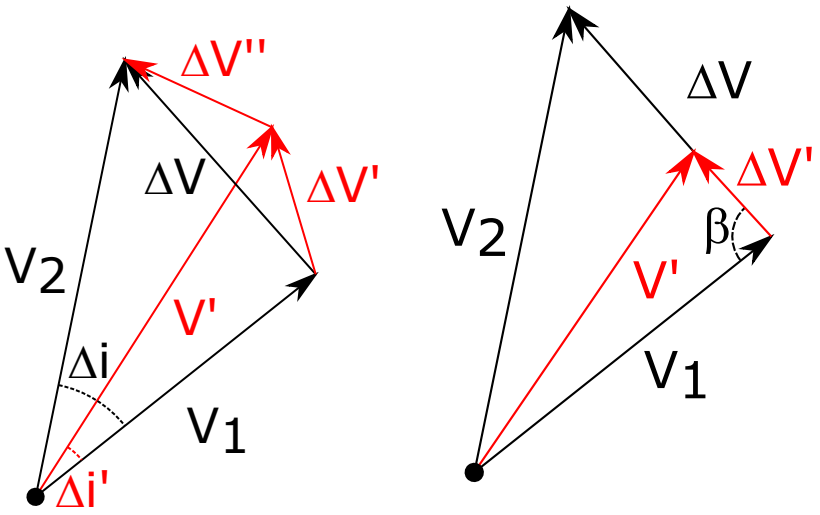
For the sake of notation, $k = 1$ is associated to the impulse with the minimum value of V_1, depicted as $V_{1,min}$. Also, for a given value of λ, the minimum aggregated Δi is obtained when $\rho_k = 1$ for every impulse. Therefore, the following quantity can be defined:

$$\eta = \sum_{k \in K} \Delta i_k(V_{1,min}; \rho_k = 1). \tag{33}$$

where η represents the maximum aggregated value of Δi for these values of ρ_k, which is achieved when λ is equal to the lowest maximum of $d\Delta V/d\Delta i$ associated to one of the impulses (i.e., $V_{1,min}$). Thus, an aggregated $\Delta i < \eta$ requires all ρ_k to be equal to 1. Likewise, an aggregated $\Delta i > \eta$ requires at least one ρ_k to be equal to -1. As a result, if $\eta - \Delta i_{tot} \geqslant 0$, any solution with at least one $\rho_k = -1$ is infeasible; therefore, the optimal ρ_k^* combination is $\rho_k^* = 1$, $\forall k \in K$. In turn, if $\eta - \Delta i_{tot} \leqslant 0$, then the solution corresponding to $\rho_k = 1$ for every impulse is infeasible. In order to determine which ρ_k^* is equal to -1 for this case, it is useful to regard $d\Delta V_k/d\Delta i_k$ as the marginal ΔV_k cost of increasing the the inclination an amount Δi_k by means of an impulse k. Hence, when the aggregated Δi to be distributed is infinitesimally greater than η, the marginal cost of increasing Δi_1 has a decreasing trend, unlike the cost associated to the rest of impulses. Thus, in that situation, the optimal allocation of an increasing quantity of the aggregated Δi requires increasing Δi_1, while decreasing the Δi_k associated to the rest of impulses. Therefore, this implies that the optimal ρ_k^* combination is the following:

$$\rho_1^* = -1, \qquad \rho_k^* = 1, \quad \forall k \in K : k \neq 1. \tag{34}$$

This expression holds true for as long as the marginal cost associated to any of the other impulses does not fall below the cost of Δi_1 for an interval long enough so as to compensate for the previous greater



(a) Non-optimal configuration (b) Optimal configuration

Figure 3: Geometry of velocity vectors on the rectifying plane for the change of inclination with subimpulses.

cost. As a result, each particular problem has to be analyzed to ascertain the range of validity of the aforementioned expression and to choose the correct ρ_k^* values when Eq. (34) no longer holds true.

Once the general solution for an arbitrary number of impulses has been introduced, we can discuss the relation between the optimal two-impulse transfer, and the case where each of these two impulses are divided into a series of *subimpulses*. Fig. 3a illustrates the geometry described by the orbital velocity vectors of a two-impulse transfer on the rectifying plane at the orbital location where one of the two impulses is performed. The orbital velocity vectors before and after the impulse, of magnitude V_1 and V_2 respectively, are indicated in black, along with the magnitude of the impulse ΔV that completes the velocity triangle. The Δi change performed at this impulse results from the resolution of the problem with the previously described methodology. Now, assume this impulse is decomposed into two subimpulses of magnitude $\Delta V'$ and $\Delta V''$, respectively (indicated in red), and such that the orbital velocity after the first subimpulse is V', which corresponds to an change of inclination of $\Delta i'$.

It is clear from the geometry of the velocity vectors that the minimum total ΔV to accomplish the desired inclination change, Δi, is accomplished when the velocity and inclination changes provided by the first subimpulse are such that the velocity increments $\Delta V'$ and $\Delta V''$ are both parallel to the original ΔV vector; as a result, the aggregated optimal ΔV consumed by the two subimpulses is identical to the one consumed by the original impulse. Such geometry is depicted in Fig. 3b, where the angle β is seen to be preserved between

the original impulse and the first subimpulse. Using the law of sines yields:

$$\sin\beta = \frac{V_2 \sin(\Delta i)}{\Delta V} = \frac{V' \sin(\Delta i')}{\Delta V'} \tag{35}$$

and from Eq. (30) the dual variable can be easily obtained for both, the original impulse (denoted as λ) and the subimpulse case (denoted as λ'):

$$\lambda = \frac{V_1 V_2 \sin\Delta i}{\Delta V} \tag{36a}$$

$$\lambda' = \frac{V_1 V' \sin\Delta i'}{\Delta V'}\,. \tag{36b}$$

Finally, substituting Eq. (35) into Eqs. (36) provides:

$$\lambda = \lambda' = V_1 \sin\beta\,. \tag{37}$$

Therefore, λ is conserved and, given the solution of the two-impulse problem, $\Delta i'$ can be computed replacing V_2 by V' in Eq. (31) while using the values of λ from the two-impulse problem along with the suitable values of ρ_k^*.

A simple numerical example is subsequently discussed. Specifically, it involves the geostationary transfer example described in [62], which comprises a transfer between two circular orbits of radii 7000 km and 42166 km, respectively, and a Δi_{tot} of 28.5 deg. First, the example is solved with a straightforward two-impulse transfer. After that, it is considered that the operational constraints of such transfer require the spacecraft to spend a certain amount of time in an intermediate orbit with a semimajor axis of 8500 km. This requirement can be achieved by simply decomposing the first impulse of the two-impulse transfer into two subimpulses.

The solution of the optimal two-impulse transfer, characterized by the impulses termed as 1/2 (i.e., first impulse out of two) and 2/2 (i.e., second impulse out of two), is summarized in Table 4. Fig. 4 depicts the optimal λ, along with the functions $d\Delta V/d\Delta i$ as a function of Δi as obtained from Eq. (30) for each of the impulses. It turns out that $\eta - \Delta i_{tot} \geqslant 0$, so the optimal ρ_k^* combination is $\rho_k^* = 1,\ \forall k \in K$. Solving Eq. (32) for that ρ_k^* combination results in $\lambda = 1.26643 \cdot 10^3\ \mathrm{m\,s^{-1}}$. However, it must be noted that the value that Eq. (30) provides for the impulse 2/2 is always less than, or equal to the rest of impulses; this implies that for every value of Δi_{tot} such that $\eta - \Delta i_{tot} \leqslant 0$, ρ_k^* would be defined by Eq. (34).

For the three-impulse case, the fist impulse is subdivided into two subimpulses, namely the impulses 1/3 (i.e., first impulse out of three) and 2/3 (i.e., second impulse out of three); the third impulse of this problem, 3/3, is thus identical to the impulse 2/2. Just like in the two-impulse problem, $\eta - \Delta i_{tot} \geqslant 0$ and the optimal ρ_k^* combination is

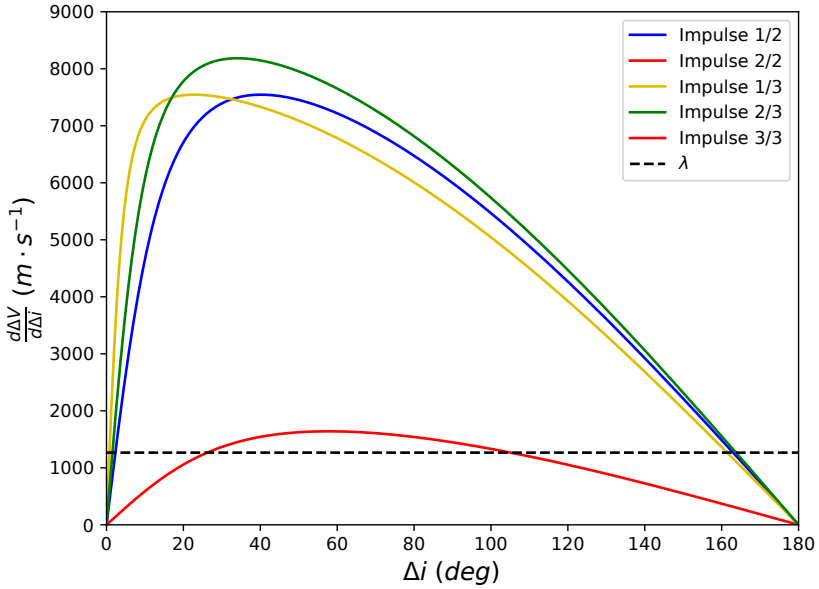


Figure 4: Solution of the multi-impulse combined maneuver

$\rho_k^* = 1,\ \forall k \in K$. The values of ρ_k^* and λ are substituted into Eq. (31), resulting in the solution shown in Table 4. Eq. (30) for the impulses 1/3 and 2/3 is also displayed in Fig. 4.

It can be seen that $d\Delta V/d\Delta i$ behaves as previously stated. In particular, it vanishes for $\Delta i = \{0, \pi\}$ and has a maximum for $\Delta i = \arccos(V_1/V_2)$, which yields a value of $d\Delta V/d\Delta i = V_1$. Consequently, as the impulses 1/2 and 1/3 have the same V_1 but a different V_2, they achieve the same maximum value but for different values of Δi.

2.4 APPLICATION TO MISSION SCENARIOS

The practical application examples discussed in Section 2.3 are recast in this section into two realistic mission scenarios of practical relevance. In both cases the optimization problem has been approached with the dual-based method and reduced to an algebraic equation equivalent to Eq. (7), which is then solved numerically using Brent's method [24].

2.4.1 *Geostationary transfer orbit with a phasing maneuver*

Due to the high demand of the geostationary orbit, it is divided into a series of slots. Consequently, the satellites that operate in such orbit are assigned to a slot, such that more than one satellite can be allocated to a single slot [26]. Hence, when performing a geostationary transfer, it is of great importance to consider the position in which

Table 4: Numerical example for the multi-impulse combined maneuver case.

Type of transfer	Impulse	ΔV [m s^{-1}]	Δi [deg]
Two-impulse	**1/2**	2362	2.30
	2/2	1759	26.20
	Total	4121	28.50
Three-impulse	**1/3**	647	0.76
	2/3	1715	1.54
	3/3	1759	26.20
	Total	4121	28.50

the geostationary orbit is reached. Therefore, this first practical case considers the optimization of a combined geostationary transfer with a phasing maneuver.

Possibly the most straightforward application of the subimpulse problem studied in Section 2.3.3 is precisely to split one of the impulses into two subimpulses, thus providing an additional impulse that can be used to simultaneously accomplish a phasing maneuver without additional ΔV consumption. In particular, the problem at hand involves four different orbits, namely, an initial orbit, the final geostationary orbit, a transfer orbit and a phasing orbit. At the beginning, the satellite will be coasting in the initial orbit. After an impulse, it will achieve the phasing orbit, where it will remain for an integer number of revolutions. Then, it will perform a subsequent impulse to achieve the transfer orbit and, after half an orbital period, another impulse will inject the satellite into the correct slot within the final orbit.

To solve this problem, it is necessary to analyze the transfer geometry resulting from the application of the aforementioned three-impulse transfer strategy. The first step to perform such analysis is to define the position of the target slot as a function of the arrival time, which can be computed as follows:

$$u_{tgt} = u^0_{tgt} + n_{tgt} t_f \,, \tag{38}$$

where u^0_{tgt} and u_{tgt} represent the argument of latitude of the target slot at the initial and arrival time, respectively, n_{tgt} is the mean motion of the target orbit and t_f is the arrival time.

The description of the three-impulse maneuver entails an arrival time of the following form:

$$t_f = t_{cst} + CT_{pha} + \frac{T_{tra}}{2} \,, \tag{39}$$

where t_{cst} is the time spent during the coasting phase, C is the number of revolutions spent in the phasing orbit, T_{pha} is the period of the phasing orbit and T_{tra} is the period of the transfer orbit.

Likewise, the angular position of the satellite at the arrival time can be derived from the geometry of the maneuver:

$$u_{sat} = u_{sat}^0 + n_{ini} t_{cst} + \pi \,, \tag{40}$$

where u_{sat}^0 and u_{sat} represent the argument of latitude of the satellite at the initial and arrival time, respectively, and n_{ini} is the mean motion of the initial orbit.

Moreover, the inclination changes have to be performed at the line of nodes. Hence, t_{cst} has to be chosen such that the satellite is located at one of the nodes when the first impulse is performed, as imposed by the following expression:

$$t_{cst} = \frac{\pi K - u_{sat}^0}{n_{ini}} \,, \tag{41}$$

where K is a natural number that defines the set of feasible values of t_{cst}. In particular, for odd values of K, the satellite will perform the first impulse at the descending node. Likewise, the satellite will perform the first impulse at the ascending node for even values of K.

It can be seen that the design variables that have to be determined to reach the target slot are T_{pha}, C and K. Therefore, the next step is to formulate an analytical condition that relates these design variables. The following parameter can be defined to facilitate that process:

$$\Delta M' = u_{tgt}^0 - \pi + n_{tgt} \left(\frac{T_{tra}}{2} - \frac{u_{sat}^0}{n_{ini}} \right) \,, \tag{42}$$

where $\Delta M'$ is the aforementioned parameter. It provides a measurement of the initial relative position between the satellite and the target slot, but corrected to take into account the effects in the angular position produced by the different phases of the maneuver.

A feasible solution of the problem at hand entails that the angular position of the satellite and the target slot have to be congruent modulo 2π at the arrival time, which can be modelled as:

$$u_{tgt} = u_{sat} + 2\pi\kappa \,, \tag{43}$$

where κ is an integer variable that imposes such congruence.

It has to be noted that the integer variables K and κ can be unified into a single variable without loss of generality. Such combination can be carried out as follows:

$$\pi K' = \pi K + 2\pi\kappa \,, \tag{44}$$

where K' is the new design variable that substitutes K and κ.

Substituting Eqs. (38-42,44) into Eq. (43) makes it possible to determine K' as a function of the rest of the design variables:

$$K' = n_{ini} \frac{\Delta M' + n_{tgt} C T_{pha}}{\pi \left(n_{ini} - n_{tgt} \right)} . \tag{45}$$

In addition, it has to be noted that, if the first impulse is the one being split into two subimpulses, T_{pha} can take any value between the periods of the initial orbit and the transfer orbit. Likewise, if the second impulse is the one being decomposed, T_{pha} can take any value between the periods of the transfer orbit and the final orbit. However, in the latter case, the phasing orbit would intersect the final orbit at each revolution, hence generating a potential risk of collision with the objects operating in geostationary orbit. Thus, the first impulse is the one being split and the corresponding range of values of the period of the phasing orbit is $T_{pha} \in [T_{ini}, T_{tra}]$, where T_{ini} is the period of the initial orbit.

All in all, the process to compute the optimal three-impulse transfer is as follows:

1. Chose an integer value of K'.
2. Isolate C in Eq. (45) and substitute the chosen K' and the extremes of the range of feasible values of T_{pha} into two instances of that equation. This provides an interval of values of C. The integers within that interval are its feasible values.
3. If there are not integer values within that interval, go back to step 1 and choose a new value for K'. Otherwise, assign to C an integer value contained in such interval.
4. T_{pha} is then obtained by means of isolating it in Eq. (45) and substituting the selected values of K' and C.
5. Compute the semimajor axis of the phasing orbit from T_{pha}.
6. Known the semimajor axis of the phasing orbit, use the dual-based methodology to determine the optimal Δi distribution among the three impulses, as explained in Section 2.3.3.

Just like in the numerical example within Section 2.3.3, the geostationary transfer example described in [62] has been used to set up this case, which involves a transfer between two circular orbits of radii 7000 km and 42166 km, respectively, and a Δi_{tot} of 28.5 deg. Moreover, both the satellite that is performing the transfer and the target slot in geostationary orbit have an initial argument of latitude of 0 deg. The application of the aforementioned algorithm to this mission scenario yields:

- $t_f = 86170$ s.

- $K' = 1$.
- $K = 1$.
- $\kappa = 0$.
- $C = 2$.
- $T_{pha} = 32038$ s.
- Semimajor axis of the phasing orbit: 21802359 m.

Finally, the three-impulse maneuver has been optimized, resulting in the solution depicted in Table 5.

Table 5: Numerical example for the geostationary transfer case.

Type of transfer	Impulse	ΔV [m s^{-1}]	Δi [deg]
Two-impulse	**1/2**	2362	2.30
	2/2	1759	26.20
	Total	4121	28.50
Three-impulse	**1/3**	2256	2.22
	2/3	106	0.08
	3/3	1759	26.20
	Total	4121	28.50

2.4.2 *Constellation multi-target rendezvous*

The proliferation of large constellations has risen the interest in on-orbit services such as the removal of defunct satellites [47, 57, 70] or satellite refueling [96, 109, 111]. Both situations demand for precise calculation of rendezvous maneuvers and careful planning of optimal rendezvous sequences. Therefore, this second practical case considers a multi-target rendezvous of one servicing spacecraft with several satellites in a constellation.

Both the drifting problem and the phasing applications studied in Sections 2.3.1 and 2.3.2, respectively, intended to optimally distribute a predefined ΔV_{tot} budget among a set of maneuvers so as to minimize the total Δt associated to the overall duration of the orbital transfers. Consequently, both applications can be integrated into a single practical case. In this case, a single servicing spacecraft has to rendezvous with several satellites situated within a constellation.

The considered constellation comprises several spacecraft in identical, circular orbits distributed on orbital planes shifted in their right

ascension of the ascending node, Ω. Each of these orbits contains several satellites shifted in their mean anomaly, M. This case has been inspired and built upon a shell of the Starlink constellation [29], in which the satellites are located at an altitude of 550 km and an inclination of 53 deg.

Specifically, this mission scenario considers three different orbital planes, each of them containing three defunct satellites. The situation of each of the planes at the initial mission time is determined by their right ascension of the ascending node, while the position of the defunct satellites within their orbital plane is represented by their initial argument of latitude. Table 6 specifies such initial positions, where the planes are ordered according to the sequence in which they are visited and u_i represents the initial argument of latitude of the i-th object to be visited within its corresponding orbital plane.

Table 6: Initial geometry of the multi-target rendezvous case.

Plane	Ω [deg]	u_1 [deg]	u_2 [deg]	u_3 [deg]
1	36.32	21.37	119.55	135.91
2	46.32	65.46	98.18	179.99
3	51.32	34.29	99.74	148.83

In addition, a ΔV budget of 100 $m\,s^{-1}$ is considered. It has to be noted that the servicing satellite is expected to spend several months in each of the drifting orbits. For such long transfers, an arbitrary final argument of latitude can be achieved with a very small change of the semimajor axis of the drifting orbit, thus requiring a negligible amount of additional ΔV. Hence, it can be assumed that, after each drifting phase, the satellite reaches the target plane in conjunction with the first of its corresponding objects. Moreover, it is assumed that the launcher is able to inject the satellite into an arbitrary orbit. As a result, the servicing satellite is considered to start the mission in conjunction with the first concerning object within the first orbital plane. Therefore, the problem at hand can be modelled as the optimal allocation of the ΔV budget among two drifting orbits (i.e., from Plane 1 to Plane 2 and from Plane 2 to Plane 3) and six phasing orbits (i.e., from u_1 to u_2 and from u_2 to u_3, for the three orbital planes).

Table 7: Multi-target rendezvous case results: drifting orbits.

$\Delta\Omega$ [deg]	ΔV [$m\,s^{-1}$]	Δi [deg]	Δt [s]
5	37.32	$1.409 \cdot 10^{-1}$	$2.9423180 \cdot 10^{7}$
10	52.78	$1.992 \cdot 10^{-1}$	$4.1594779 \cdot 10^{7}$

Table 8: Multi-target rendezvous case results: phasing orbits.

ΔM [deg]	ΔV [$m\,s^{-1}$]	a [m]	Δt [s]	ΔV Corr. [$m\,s^{-1}$]	a Corr. [m]	Δt Corr. [s]
16.36	0.91	$6.920177 \cdot 10^6$	$1.442049 \cdot 10^6$	$-1.07 \cdot 10^{-3}$	$9.73 \cdot 10^{-1}$	$1.69 \cdot 10^3$
32.72	1.29	$6.919832 \cdot 10^6$	$2.039420 \cdot 10^6$	$6.83 \cdot 10^{-6}$	$-6.23 \cdot 10^{-3}$	$-1.08 \cdot 10^1$
49.09	1.58	$6.919567 \cdot 10^6$	$2.497778 \cdot 10^6$	$1.39 \cdot 10^{-4}$	$-1.27 \cdot 10^{-1}$	$-2.19 \cdot 10^2$
65.45	1.83	$6.919344 \cdot 10^6$	$2.884196 \cdot 10^6$	$1.89 \cdot 10^{-3}$	$-1.72 \cdot 10^0$	$-2.98 \cdot 10^3$
81.81	2.04	$6.919147 \cdot 10^6$	$3.224631 \cdot 10^6$	$-8.58 \cdot 10^{-5}$	$7.82 \cdot 10^{-2}$	$1.35 \cdot 10^2$
98.18	2.24	$6.918969 \cdot 10^6$	$3.532419 \cdot 10^6$	$-9.59 \cdot 10^{-4}$	$8.74 \cdot 10^{-1}$	$1.51 \cdot 10^3$

Table 7 shows the $\Delta\Omega$ associated to each drifting orbit, as well as the obtained solution. In turn, Table 8 does the same with the ΔM associated to each phasing orbit. It can be seen that the majority of ΔV has been assigned to modify the inclination of the drifting orbits; this is expected since, for any given ΔV value, the marginal cost of assigning additional ΔV to the drifting orbits is much lower than for the phasing orbits, as can be appreciated from Fig. 5. Table 8 also depicts the corrections applied to the ΔV, a and Δt associated to each phasing orbit so that an integer number of revolutions is performed (i.e., the solution of the combinatorial optimization problem defined by Eqs. (25)). These ΔV and a corrections are remarkably small and the aggregated Δt correction amounts to around 128 s, which implies that the solution directly obtained by the dual-based optimization method, though infeasible, represents a good estimation of the optimal feasible solution.

Fig. 5 shows the optimal λ, along with the functions $d\Delta t/d\Delta V$ as a function of ΔV, as obtained from Eqs. (12,19) for each of the concerning orbits. It can bee seen that every single one of these functions is strictly monotonically increasing, thus the obtained solution is a global optimum and the inverse functions of Eqs. (12,19) can be easily computed because of the bijective relation between them.

2.5 CONCLUSIONS

This manuscript proposes a novel optimization method for single-constraint Nonlinear Programming problems with a specific structure. Its main advantage is the ability to transform an optimization problem with an arbitrary number of variables into a root-finding problem of a univariate algebraic equation.

Three practical application examples of this methodology, based upon typical impulsive maneuver optimization problems, have been introduced and developed in detail to illustrate the application of

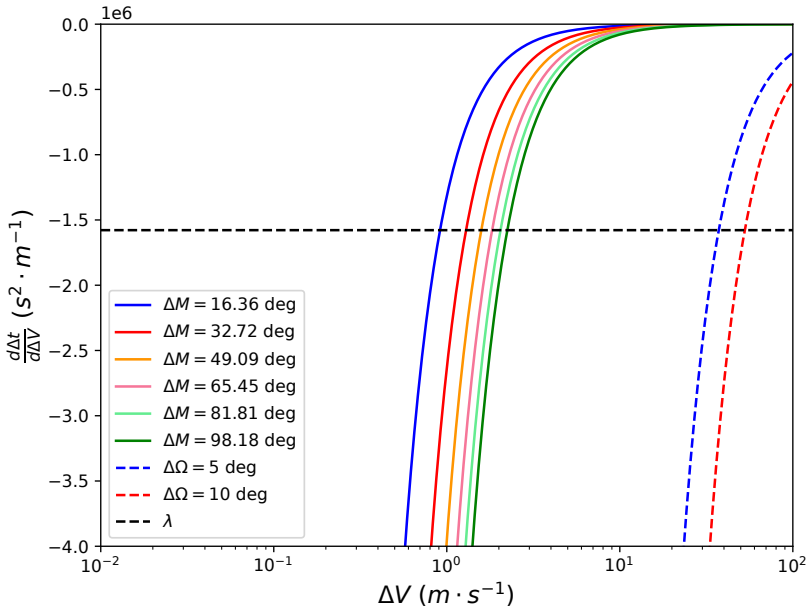


Figure 5: Multi-target rendezvous solution

this methodology. First, the optimization of inclination changes for sequences of drifting orbits has been addressed. Second, the optimization of the semi-major axes for sequences of phasing orbits has been discussed; since the resolution of the latter yields phasing maneuvers that provide fractional numbers of revolutions, an efficient correction method has been devised, which overcomes this issue by providing optimal solutions for a integer numbers of revolutions on the phasing orbit. In the third place, the optimization of combined, multi-impulse maneuvers has been explored, where the dual-based methodology has provided an insightful approach to demonstrate that the value of the only dual variable of the problem is conserved when splitting impulses into several *subimpulses*.

Finally, the utility of the proposed methodology has been assessed by solving two more complex practical cases related to potential mission scenarios. On the one hand, the conservation of the dual variable λ has been exploited to compute the optimal Δi distribution of a three-impulse geostationary transfer orbit combined with a phasing maneuver, thus readily performing a phasing maneuver without additional ΔV consumption. On the other hand, a multi-target rendezvous problem for satellites distributed in different orbital planes of a constellation has been solved by simultaneously optimizing the values of the inclinations of two drifting orbits, along with the semi-major axes of six phasing orbits as a function of a single decision variable.

All in all, these examples illustrate the potential of the proposed dual-based methodology for practical optimization problems involv-

ing impulsive maneuvers. It has to be noted that all the aforementioned examples constitute nonconvex NLP problems. Therefore, general NLP solvers are not able to prove the global optimality of their solution and might even fail to converge to a local optimum. In turn, the proposed methodology has been able to readily obtain their global optima.

Regarding the minimum-time orbit phasing and drift examples, the obtention of the global optimum is guaranteed because the functions $d\Delta t/d\Delta V$ are monotonic, thus resulting in a unique solution that fulfills the conservation of the dual variable λ. Regarding the multi-impulse combined maneuvers examples, the global optimum is obtained according to the analysis of $d\Delta V/d\Delta i$ described in Section 2.3.3, where the fact that such functions have a single maximum is exploited. A future line of work will consider problems involving df_k/dg_k functions with more complex structures (i.e., non monotonic and with multiple maxima and minima), so as to provide a more general discussion about the obtention of their global optima.

An additional future line of work will explore the possibility of extending the methodology to deal with functions f_k and g_k of more than one variable. This is of particular interest for the minimum-time orbital drift example, as it would allow to decide, not only the inclination, but also the semimajor axis of the drifting orbit. Moreover, this would open the possibility to integrate that example with the multi-impulse combined maneuver example, which would allow to achieve the correct phasing at the arrival in the target orbit (i.e., after the drift). Interestingly, the constellation multi-target rendezvous example assumed that the spacecraft reached the target orbits with the correct phasing and neglected the ΔV necessary to achieve it. Therefore, this extension would guarantee that this assumption is true and would allow to enhance the fidelity of the solution of this example.

3

RELATIVE INCLINATION STRATEGY FOR LOW-THRUST TRANSFERS

3.1 INTRODUCTION

The high concentration of space debris in particular zones of Low Earth Orbit (LEO) constitutes a considerable risk for the future of space operations. Specifically, it enhances the likelihood of a collision cascade phenomenon that would produce an uncontrollable generation of debris fragments [65]. Further studies have determined that the active removal of hazardous pieces of debris is necessary to achieve the stabilization of the number of objects within the zones of interest [71, 75]. In particular, it is expected that Active Debris Removal (ADR) missions will be able to remove several objects with a single servicing spacecraft. The use of low thrust propulsion constitutes a promising approach to ameliorate the substantial fuel requirements of such multi-target missions. Zuiani and Vasile [115] developed a first-order solution of the perturbed Keplerian motion to model low thrust transfers for multi-target ADR missions. This approach was later extended with the introduction of the effect of the J_2 perturbation as well as the atmospheric drag by means of the use of asymptotic analytical solutions of the Keplerian motion [40].

The most relevant candidate objects in LEO describe near-circular orbits. Thus, it is reasonable to consider transfers between circular orbits during the preliminary mission design. The classic Edelbaum approach [41] makes use of an averaged orbital dynamics to obtain an analytical solution of a minimum-time continuous-thrust transfer between two circular orbits. The Edelbaum solution was later reformulated as an optimal control problem [62] and extended to consider richer transfer models [27, 50, 66].

Moreover, such candidate objects are concentrated among different clusters of orbits with similar altitude and inclination. However, the objects within a cluster are, in general, arbitrarily spread in right ascension of the ascending node (RAAN). The usual strategy to traverse large RAAN differences during a transfer is to leverage the RAAN drift produced by the J_2 perturbation. To that end, Cerf [32] formulates an optimal control problem that extends the Edelbaum problem by means of considering the magnitude of the thrust acceleration as an additional control variable. Specifically, this problem considers tangential and out-of-plane thrust with a reversal of the thrust direction in the antinodes of the orbit. Hence, the control variables of the problem are the thrust yaw angle and the magnitude of the thrust acceler-

ation, which are exclusively used to modify the semimajor axis and inclination of the orbit. In turn, the whole RAAN difference between the initial and target orbits is overcome by the effect of the J_2 perturbation. It is demonstrated that the optimal control law of the thrust magnitude follows a thrust-coast-thrust profile with maximum thrust acceleration during the propulsive arcs. Thus, the optimal solution of this problem uses the first thrust arc to increase the nodal precession, which is exploited during the coasting arc. Finally, another thrust arc achieves the terminal values of the semimajor axis and inclination. An indirect optimization method is proposed to solve this problem.

Wen et al. [104] extend the Cerf approach by introducing the orbital position of the thrust reversal as a control variable. This new variable determines the effect of the thrust in the inclination and RAAN. In particular, if the thrust reversal is made at the antinodes of the orbit, only the inclination will be modified. In turn, if it is performed at the orbital nodes, only the RAAN will be changed. Hence, intermediate values of the new control variable result in different proportions of the influence of the thrust in said orbital parameters. This new approach, referred to as Yaw Switch Steering (YSS), is able to achieve significant improvements in fuel consumption (with respect to the Cerf approach) for time-constrained transfers, especially for restrictive values of such maximum time.

The optimal target selection for ADR missions might require the consideration of pools of hundreds of relevant objects [8, 74], as well as the computation of a substantial amount of possible transfers. Hence the importance of developing fast and reliable evaluation methods for such transfers. In addition to the Cerf approach, Ref. [32] presents another relevant method, referred to as the Split Edelbaum Strategy (SES). In particular, it is a simplification of the Cerf approach in which the thrust arcs are independently computed using the Edelbaum analytical solution. Hence, the Δv of the propulsive arcs is analytically computed, while the RAAN variation produced by the J_2 perturbation during those arcs is obtained by numerically integrating its corresponding differential equation. The merging of the coasting and propulsive arcs results in a two-variable Nonlinear Programming problem (NLP). This method is presented as a tool to obtain initial guesses for the Cerf approach. However, it is of remarkable relevance for the formulation of large-scale combinatorial problems because of its accuracy and computational efficiency.

More recently, Shen [94] developed two explicit analytical approximations for the aforementioned three-arc transfers. On the one hand, the Shen(a,i) solution considers that the whole RAAN is exclusively modified by the effect of the J_2 perturbation, just like the Cerf approach. On the other hand, the Shen(a,i,Ω) solution also considers an active use of the thrust to modify the RAAN. It is shown that both solutions achieve a good approximation of the Δv consumed during

this kind of transfers, with the Shen(a,i,Ω) solution being marginally better. Moreover, the evaluation time of this kind of analytical approximations is potentially several orders of magnitude smaller than the resolution time of other numerical methods.

All in all, the possibility of directly modifying the RAAN for the aforementioned kind of transfers seems like a promising approach that merits further investigation. As the YSS approach provides a high-quality solution at a high computational cost and the Shen methodology provides a remarkably computationally-efficient approximation of the Δv consumption, the question is if it is possible to devise a methodology that exploits the direct RAAN modification concept, while achieving a trade-off between solution quality and computational efficiency.

This work proposes a methodology that fulfills such purpose by means of taking advantage of maneuvers that modify the relative inclination with respect to the target orbit (i.e., the angle of the current orbital plane with the target orbital plane). The first step to configure such methodology is to devise a computationally-efficient technique to solve continuous-thrust transfers that modify the relative inclination. To that end, this work adds the effect of the J_2 perturbation to the Edelbaum approach, while considering the semimajor axis and relative inclination as the state variables (as opposed to the Cerf and YSS approaches, that need to introduce an additional state variable into the Edelbaum model in order to consider the effect of the J_2 perturbation). The optimal solution of this problem, albeit not analytical, only requires the determination of the initial value of the thrust yaw angle.

Additionally, two heuristic control laws for such continuous-thrust transfers are defined to further accelerate the resolution time of the problem. Specifically, those suboptimal control laws are derived by ignoring some basic properties of the optimal solution and have been empirically chosen because they achieve solutions very similar to the optimal trajectory for short thrust arcs.

Finally, the SES methodology is a notable example of solution quality and computational efficiency trade-off. Hence, the proposed approach replaces the final thrust arc of the SES approach with the proposed relative-inclination continuous-thrust transfer. This results in a three-variable NLP problem. Therefore, this methodology is able to exploit the RAAN change executed by the thrust to obtain potential Δv improvements, while maintaining a computational complexity similar to the one of the SES approach.

The remainder of this manuscript is organized as follows. Section 3.2 describes the resolution method of the continuous-thrust optimal control problem. Section 3.3 defines the three-arc transfer NLP problem. Section 3.4 shows numerical experiments of the proposed methodol-

ogy and provides a comparison with other relevant approaches. Finally, Section 3.5 summarizes the main conclusions of this work.

3.2 CONTINUOUS-THRUST TRANSFER

The problem at hand is to provide a time-optimal solution for a particular class of continuous-thrust transfers between circular orbits. Specifically, the proposed maneuver involves the application of tangential and out-of-plane thrust with the thrust yaw angle as the sole control variable. Moreover, the out-of-plane thrust is reversed when traversing the antinodes of the relative line of nodes with respect to the target orbit. Thus, the considered thrust produces changes in the semimajor axis and in the relative inclination with respect to the target orbit. Additionally, the effect of the average J_2 perturbation in the RAAN is also considered.

3.2.1 *Problem dynamics*

The Edelbaum analytical solution is equally valid to represent inclination changes as well as changes of relative inclination with respect to the target orbit. The reason for it is that both kinds of maneuvers are geometrically identical, but seen from reference frames with different orientation, that is, the usual equatorial coordinate system and a reference frame such that its XY plane is the target plane and its Z axis goes along the angular momentum of the target orbit, respectively. However, this is not true when considering the effect of the J_2 perturbation, as its contribution is not invariant under reference frame rotations. Thus, the dynamics equations that model the problem at hand are equivalent to the ones in [62] but with an additional term (γ) that quantifies the effect of the RAAN drift on the relative inclination. In particular, they represent the average rate of change per orbital revolution of the semimajor axis and relative inclination under the effect of tangential and out-of-plane thrust and the averaged J_2 perturbation. It is assumed that such averaged orbit will remain circular throughout the whole transfer. Moreover, the thrust will keep a constant yaw angle per revolution, with a thrust reversal performed at the antinodes corresponding to the relative line of nodes with respect to the target orbit, thus resulting in the variation of the relative inclination with respect to such orbit (instead of the usual inclination defined with respect to the equatorial plane). Finally, the thrust acceleration magnitude is considered to remain constant throughout the whole transfer. All in all, the dynamics equations are defined as follows:

$$\frac{da}{dt} = 2\sqrt{\frac{a^3}{\mu}} f \cos\beta \tag{46}$$

$$\frac{di^*}{dt} = \frac{2}{\pi}\sqrt{\frac{a}{\mu}} f \sin\beta + \gamma \tag{47}$$

where a and i^* are the semimajor axis and the relative inclination of the current orbit, respectively; f is the thrust acceleration; β is the thrust yaw angle; and μ is the gravitational parameter of the Earth.

The averaged RAAN drift produced by the J_2 perturbation for a circular orbit is:

$$\frac{d\Omega}{dt} = -\frac{3}{2} R_\oplus^2 J_2 \sqrt{\frac{\mu}{a^7}} \cos i \tag{48}$$

where Ω is the RAAN, $R_\oplus$ is the equatorial radius of the Earth, and J_2 is the coefficient of the spherical harmonic of degree 2 and order 0 of the Earth's gravity field.

Consequently, the determination of γ requires two tasks. On the one hand, the term $\cos i$ has to be redefined as a function of orbital parameters associated with the target orbit reference frame. On the other hand, the rate of change of Ω has to be turned into a rate of change of i^*.

Regarding the first task, the ascending nodes of the current (N) and target (N_f) orbits as well as the relative ascending node (N*) constitute the spherical triangle shown in Fig. 6, resulting in the following trigonometry laws:

$$\cos i^* = \cos i \cos i_f + \sin i \sin i_f \cos\Delta\Omega \tag{49}$$

$$\cos i = \cos i_f \cos i^* - \cos\varpi \sin i_f \sin i^* \tag{50}$$

$$\frac{\sin i^*}{\sin\Delta\Omega} = \frac{\sin i}{\sin\varpi} \tag{51}$$

where $\Delta\Omega = \Omega_f - \Omega$; $\varpi = \omega_f + \Omega^*$; Ω^* is the RAAN associated with the target orbit reference frame; and i_f, Ω_f, and ω_f are the inclination, RAAN, and argument of perigee of the target orbit, respectively.

Regarding the second task, as γ quantifies the rate of change of i^* strictly produced by the RAAN drift of the current and target orbits, the subsequent chain rule can be applied:

$$\gamma = \left.\frac{di^*}{dt}\right|_{J_2} = \frac{\partial i^*}{\partial\Delta\Omega}\frac{d\Delta\Omega}{dt} \tag{52}$$

Taking the corresponding partial derivative of Eq. (49) and substituting Eq. (51) yields:

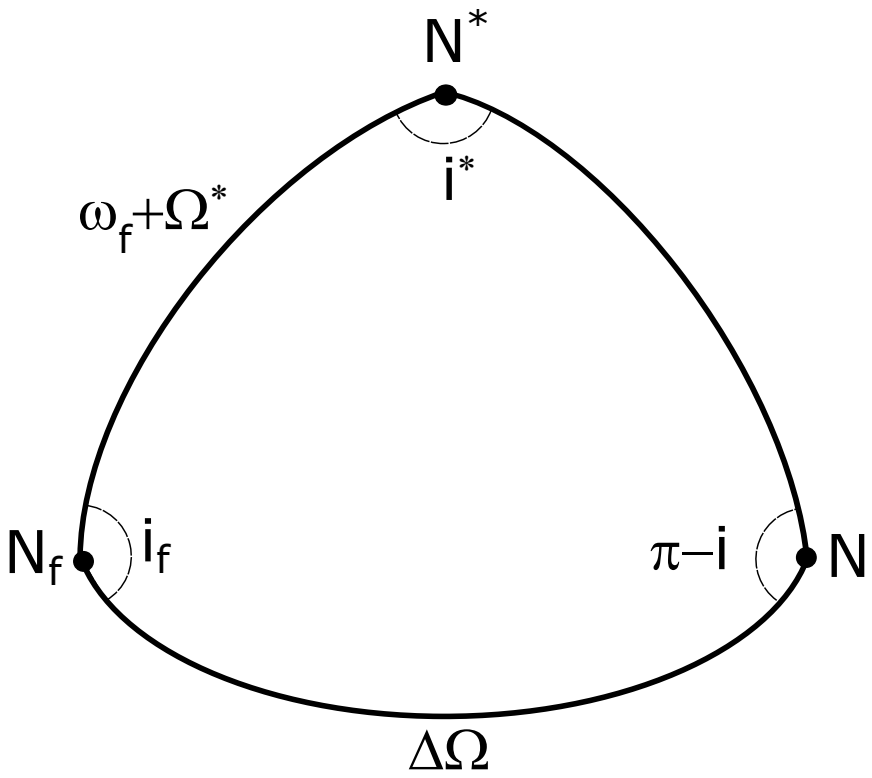


Figure 6: Spherical triangle formed by ascending nodes.

$$\frac{\partial i^*}{\partial \Delta\Omega} = \sin i_f \sin \varpi \tag{53}$$

In turn, subtracting the RAAN drifts experienced by the target and the current orbits [Eq. (48)] and substituting Eq. (50) results in:

$$\frac{d\Delta\Omega}{dt} = -\frac{3}{2} R_\oplus^2 J_2 \left(\sqrt{\frac{\mu}{a_f^7}} \cos i_f - \sqrt{\frac{\mu}{a^7}} \left(\cos i_f \cos i^* - \cos \varpi \sin i_f \sin i^* \right) \right) \tag{54}$$

It has to be noted that the term ϖ appears in the definition of γ, but the differential equation that describes its temporal evolution is not considered in the problem dynamics. Therefore, given the value of γ at a certain orbital revolution, the propagation of Eqs. (46) and (47) is not enough to determine its value at the subsequent one. Fortunately, the following properties of the considered transfer can be exploited to propagate γ without the need for adding an additional differential equation to the problem dynamics:

1. i is modified exclusively by the thrust.
2. ϖ is modified exclusively by the J_2 perturbation.

3. $\Delta\Omega$ is modified by both the thrust and the J_2 perturbation.

Thus, given the averaged values of all the variables for the current orbital revolution (i.e., a, i^*, i, $\Delta\Omega$, ϖ, β, and γ), their propagation to the subsequent one can be performed as follows:

1. The values of a and i^* can be obtained with a straightforward propagation of Eqs. (46) and (47). Specifically, substituting a, i^*, β, and γ into those equations gives the average rates of change of a and i^* during the current revolution. Then, multiplying those rates of change by the orbital period and adding the resulting quantities to the current values of a and i^* results in the desired averaged values for the subsequent revolution. This integration scheme is analogously used for the propagation of all the differential equations considered during the resolution of this problem.

2. When considering Keplerian dynamics, ϖ remains constant, while the obtained i would be identical to the one of the J_2-perturbed case. Consequently, computing i^* for the Keplerian case (i.e., with $\gamma = 0$) and introducing it, together with ϖ, into Eq. (50) would result in the desired value of i.

3. The Keplerian value of $\Delta\Omega$ results from substituting the Keplerian values of i, ϖ, and i^* in Eqs. (49) and (51). Then, its corresponding nodal precession [Eq. (48)] can be added to obtain its J_2-perturbed counterpart.

4. Subsequently, the J_2-perturbed value of ϖ results from Eqs. (50) and (51).

5. Finally, Eq. (52) is used to compute γ.

3.2.2 *Optimality conditions*

The optimal control formulation of the problem at hand involves the minimization of the transfer time, as depicted in the following objective function:

$$J = \min \int_{t_0}^{t_f} dt \tag{55}$$

where t_0 is the fixed initial time and t_f is a free final time, subjected to the dynamics defined by Eqs. (46) and (47) and the following boundary conditions:

$$\begin{aligned} a(t_0) &= a_0 \quad i^*(t_0) = i_0^* \\ a(t_f) &= a_f \quad i^*(t_f) = 0 \end{aligned} \tag{56}$$

where a_0 and a_f are the initial and final values of the semimajor axis, respectively, and i_0^* is the initial relative inclination. It has to be noted that, as the thrust acceleration magnitude is considered to remain constant throughout the whole transfer, the minimization of the transfer time is equivalent to the minimization of the consumed Δv. In addition, such Δv can be computed by simply multiplying the transfer time by the thrust acceleration magnitude.

The Pontryagin's minimum principle states that the optimal solution of this problem requires the selection of a control law of β such that the following Hamiltonian function is minimized throughout the whole optimal trajectory:

$$\mathcal{H} = 1 + \lambda_a \left(2\sqrt{\frac{a^3}{\mu}} f \cos\beta \right) + \lambda_i \left(\frac{2}{\pi}\sqrt{\frac{a}{\mu}} f \sin\beta + \gamma \right) \tag{57}$$

where λ_a and λ_i are the costate variables associated to Eqs. (46) and (47), respectively. The Euler-Lagrange equations determine the rates of change of the costate variables:

$$\frac{d\lambda_a}{dt} = -\frac{\partial \mathcal{H}}{\partial a} = -\frac{f}{\sqrt{\mu}} \left(3\lambda_a \sqrt{a} \cos\beta + \frac{\lambda_i}{\pi\sqrt{a}} \sin\beta \right) - \lambda_i \frac{\partial \gamma}{\partial a} \tag{58}$$

$$\frac{d\lambda_i}{dt} = -\frac{\partial \mathcal{H}}{\partial i^*} = -\lambda_i \frac{\partial \gamma}{\partial i^*} \tag{59}$$

The necessary condition for the Hamiltonian function minimization is:

$$\frac{\partial \mathcal{H}}{\partial \beta} = -2\lambda_a \sqrt{\frac{a^3}{\mu}} f \sin\beta + \frac{2}{\pi} \lambda_i \sqrt{\frac{a}{\mu}} f \cos\beta = 0 \tag{60}$$

resulting in the following optimal control law:

$$\tan\beta = \frac{\lambda_i}{\lambda_a \pi a} \tag{61}$$

It has to be noted that time does not explicitly appear in the Hamiltonian function [Eq. (57)]. Hence, $\mathcal{H}$ is constant throughout the whole optimal trajectory. Furthermore, t_f is a free variable and its associated boundary conditions are time-independent. Consequently, $\mathcal{H}(t_f) = 0$ and, by extension, is null during the whole trajectory. This condition, along with the optimal control law [Eq. (61)], can be used to unambiguously define the sine and cosine of β as functions of the costate variables:

$$\sin\beta = -\frac{2\lambda_i\sqrt{a}f}{\pi\sqrt{\mu}\,(1+\lambda_i\gamma)} \tag{62}$$

$$\cos\beta = -\frac{2\lambda_a\sqrt{a^3}f}{\sqrt{\mu}\,(1+\lambda_i\gamma)} \tag{63}$$

The costate variables can be also given as functions of β as follows:

$$\lambda_i = -\frac{\sin\beta}{\frac{2}{\pi}\sqrt{\frac{a}{\mu}}f+\gamma\sin\beta} \tag{64}$$

$$\lambda_a = -\frac{\cos\beta}{2\sqrt{\frac{a^3}{\mu}}f+\gamma\pi a\sin\beta} \tag{65}$$

Finally, the strong Legendre–Clebsch condition guarantees that a local minimum is obtained (as opposed to a maximum or saddle point).

$$\frac{\partial^2\mathcal{H}}{\partial\beta^2} = -2\lambda_a\sqrt{\frac{a^3}{\mu}}f\cos\beta - \frac{2}{\pi}\lambda_i\sqrt{\frac{a}{\mu}}f\sin\beta > 0 \tag{66}$$

Recalling that $\mathcal{H} = 0$ and using Eq. (57) yields:

$$\lambda_i\gamma > -1 \tag{67}$$

Then, substituting λ_i from Eq. (64) gives:

$$\frac{\gamma\sin\beta}{\frac{2}{\pi}\sqrt{\frac{a}{\mu}}f+\gamma\sin\beta} < 1 \tag{68}$$

This expression implies that the signs of its denominator and the first term of such denominator have to be identical. By definition, such first term is positive. Hence, the condition to obtain a local minimum is:

$$\frac{2}{\pi}\sqrt{\frac{a}{\mu}}f+\gamma\sin\beta > 0 \tag{69}$$

3.2.3 *Heuristic control laws*

When focusing on the computational efficiency of the resolution of optimal control problems, it is worth investigating promising heuristic control laws. Specifically, control laws independent of the costates that, albeit suboptimal, can achieve near-optimal solutions under suitable circumstances. This way, such near-optimal solutions can be obtained by simply propagating the state equations to the desired final state. Two promising heuristic control laws are subsequently presented, namely, the proportional control law and the corrected Edelbaum strategy.

Such heuristic control laws have been derived by means of neglecting some basic properties of the optimal solution. In particular, the proposed control laws have been empirically chosen because they achieve solutions very similar to the optimal trajectory for short thrust arcs. Hence, if it is expected that this hypothesis will be true, such heuristic control laws should be used instead of the aforementioned optimal solution, further improving the computational efficiency while obtaining solutions very close to the optimal. However, it has to be taken into account that the quality of the heuristic solutions can be significantly deteriorated by the length of the thrust arc.

Proportional control law

The selection of β involves a trade-off between the magnitude of the in-plane (to modify a) and out-of-plane (to change i^*) thrust components. The proportional control law allocates the thrust components such that the rates of change of the state variables are proportional to their difference with the target values, that is:

$$\frac{da/dt}{di^*/dt} = \frac{a - a_f}{i^*} \tag{70}$$

For the sake of simplicity, Eq. (70) is reformulated as follows:

$$\frac{\phi_a \cos\beta}{\phi_i \sin\beta + \gamma} = d \tag{71}$$

where d is the right hand side of Eq. (70) and ϕ_a, ϕ_i are the factors that multiply $\cos\beta$, $\sin\beta$ in Eqs. (46) and (47), respectively.

Using the Pythagorean trigonometric identity, the sine and cosine of β can be isolated. This results in two possible solutions of Eq. (71). Nevertheless, one of those solutions increases the gap between both of the current states and the target orbit, while the other solution does the opposite. Thus, the only relevant solution is:

$$\sin\beta = -\frac{\phi_i\gamma d^2 + \phi_a\sqrt{\phi_a^2 + (\phi_i^2 - \gamma^2)d^2}}{\phi_a^2 + \phi_i^2 d^2} \tag{72}$$

$$\cos\beta = \frac{d\left(\phi_a\gamma - \phi_i\sqrt{\phi_a^2 + (\phi_i^2 - \gamma^2)d^2}\right)}{\phi_a^2 + \phi_i^2 d^2} \tag{73}$$

Corrected Edelbaum control law

For short thrust arcs, it is reasonable to assume that the effect of the J_2 perturbation will be very small. Thus, another promising heuristic control law can be obtained by neglecting the influence of such perturbation in the value of β. Specifically, this control law can be readily obtained by expressing the Edelbaum analytical expression of β as a function of i^* and a.

$$\beta = \operatorname{atan2}\left(-\sin\left(\frac{\pi}{2}i^*\right), \sqrt{\frac{a_f}{a}} - \cos\left(\frac{\pi}{2}i^*\right)\right) \tag{74}$$

However, neglecting the effect of the J_2 perturbation entails that the expected terminal state will not be exactly achieved. This can be solved by using the proportional control law during the last revolution of the transfer. Hence the name of corrected Edelbaum strategy.

3.2.4 *Problem resolution*

In general, the resolution of the problem at hand involves the propagation of the dynamics differential equations [Eqs. (46),(47),(58), and (59)], while fulfilling the optimal control law [Eqs. (62) and (63)]. Moreover, the accomplishment of the desired boundary conditions at t_f [Eq. (56)] requires the determination of specific initial values for the costate variables. This can be difficult because such variables do not possess a clear physical meaning. Nevertheless, in this particular case, Eqs. (64) and (65) provide the values of such costates as a function of β. Therefore, the boundary conditions at t_f can be achieved by simply selecting a suitable initial value of β, which has a bounded domain as well as a straightforward meaning. Consequently, the whole resolution process can be encapsulated in the following scalar univariate shooting function:

$$\psi : \beta(t_0) \to a(t_f) - a_f \tag{75}$$

Given an initial value of β, the shooting function propagates the dynamics equations until $i^* = 0$ and returns the defect of the resulting

semimajor axis at that epoch, i.e., t_f. Hence, a root of ψ generates an optimal trajectory that complies with the boundary conditions. The following two-stage approach has been devised to obtain the aforementioned root:

1. The initial β angles corresponding to the proportional and corrected Edelbaum heuristic control laws are used to initialize a secant method. This method is executed until two values of ψ with different signs are found or until the solution is obtained, whichever happens first.
2. If the solution has not been obtained during the previous stage, the β angles corresponding to the values of ψ with different signs are used to execute Brent's bracketing method [24]. In this way, the convergence to an optimal solution is guaranteed (assuming that ψ is continuous).

3.3 THREE-ARC TRANSFER

When dealing with transfers involving a significant $\Delta\Omega$, continuous-thrust maneuvers may require an infeasible amount of Δv. In those cases, it is interesting to exploit the RAAN drift produced by the J_2 perturbation. Specifically, the usual strategy is to introduce an intermediate coasting arc in which the satellite will remain in a drifting orbit with an advantageous nodal precession.

3.3.1 *SES strategy*

The previously mentioned SES strategy [32] comprises two propulsive arcs, modelled with the Edelbaum solution, split apart by a coasting arc. In this way, the propulsive arcs simultaneously modify a and i, while the whole $\Delta\Omega$ is overcome exclusively by the J_2 perturbation. Therefore, defining $\Delta v_{Ed}(a_0, i_0, a_1, i_1)$ as the Δv consumed during an Edelbaum transfer from (a_0, i_0) to (a_1, i_1), the Δv spent during the whole maneuver is:

$$\Delta v = \Delta v_{Ed}(a_0, i_0, a_0 + \Delta a, i_0 + \Delta i) + \Delta v_{Ed}(a_0 + \Delta a, i_0 + \Delta i, a_f, i_f) \tag{76}$$

where Δa and Δi are the necessary parameter variations to reach the drifting orbit. In turn, the Δt spent during this maneuver is:

$$\Delta t = \frac{\Delta v}{f} + t_{drift} \tag{77}$$

where t_{drift} is the drifting time, whose value is unambiguously determined by the fact that a RAAN difference of $\Delta\Omega$ has to be crossed over during the transfer. That is:

$$t_{drift} = -\frac{\Delta\Omega + \Delta(\Delta\Omega_1) + \Delta(\Delta\Omega_2)}{\Delta\dot{\Omega}_{drift}} \tag{78}$$

where $\Delta(\Delta\Omega_1)$ and $\Delta(\Delta\Omega_2)$ are the variations of $\Delta\Omega$ due to the J_2 perturbation during the first and second propulsive arcs, respectively, and $\Delta\dot{\Omega}_{drift}$ is the rate of change of $\Delta\Omega$ during the coasting arc. Both $\Delta(\Delta\Omega_1)$ and $\Delta(\Delta\Omega_2)$ are computed through a numerical integration of Eq. (48).

Eqs. (76) and (77) can be used to formulate a rather manageable single-constraint NLP problem of two variables (i.e., Δa and Δi). Hence the efficiency and reliability of this method.

3.3.2 *Relative Inclination Change (RIC) strategy*

The SES strategy proves especially Δv-efficient when the available time to perform a transfer is such that the whole $\Delta\Omega$ can be traversed with moderate variations of the RAAN drift. However, when considering short transfer times, it is notably advantageous to use the thrust to also modify the RAAN [104]. The RIC strategy is proposed to exploit this concept. Specifically, this strategy uses the same structure as the SES strategy, but substitutes the second Edelbaum propulsive arc with the relative inclination change maneuver described in Section 3.2. Hence, this approach is expected to outperform the SES strategy for tight transfer times. In turn, for longer transfer times both approaches are nearly identical, the difference being that the RIC strategy inevitably performs a small $\Delta\Omega$ change. Therefore, this approach is expected to be slightly worse than the SES strategy for such transfer times.

Defining $\Delta v_{RIC}(a_0, i_0, a_1, i_1, \Delta\Omega_0)$ as the Δv consumed during a relative inclination change transfer from (a_0, i_0) to (a_1, i_1) with an initial RAAN difference of $\Delta\Omega_0$, the Δv spent during the whole maneuver is:

$$\begin{aligned}\Delta v = {} & \Delta v_{Ed}(a_0, i_0, a_0 + \Delta a, i_0 + \Delta i) + \\ & + \Delta v_{RIC}(a_0 + \Delta a, i_0 + \Delta i, a_f, i_f, \Delta\Omega_0 + \Delta(\Delta\Omega_1) + \Delta\dot{\Omega}_{drift} t_{drift})\end{aligned} \tag{79}$$

where $\Delta\Omega_0$ is the initial RAAN difference before the first propulsive arc and $\Delta(\Delta\Omega_1)$ and $\Delta\dot{\Omega}_{drift} t_{drift}$ are the variations of $\Delta\Omega$ due to the J_2 perturbation during the first propulsive arc and the coasting arc, respectively. Therefore, the term $\Delta\Omega_0 + \Delta(\Delta\Omega_1) + \Delta\dot{\Omega}_{drift} t_{drift}$ represents the initial RAAN difference for the second propulsive arc. Then, Eq. (49) is used to compute i^* and the value of Δv_{RIC} can be obtained as described in Section 3.2.

The Δt spent during this maneuver is also defined by Eq. (77). However, as the second propulsive arc is able to modify $\Delta\Omega$, t_{drift} is a variable to be determined during the optimization. Thus, Eqs. (77) and (79) result in a NLP problem of three variables (i.e., Δa, Δi, and t_{drift}).

3.4 RESULTS

Numerical experiments have been performed to provide a comparison between the RIC strategy and other relevant methodologies. In decreasing order of complexity, the considered approaches can be classified into optimal control problems (YSS [104], Cerf [32]), NLP problems (RIC, SES [32]), and analytical approximations (Shen [94]). It is well known that the solutions obtained by indirect optimization methods, like the ones used in the Cerf and YSS aproaches, are usually very sentitive to small changes of the initial values of the costates. Specifically, both approaches require guessing the initial values of five variables, namely, three costates and the initial and final times of the coasting arc. Therefore, their resolution is computationally intensive and their convergence might be too unreliable for the formulation of large-scale combinatorial problems. However, the strong optimality conditions imposed by these methods result in high-quality and accurate solutions, hence making them especially suitable for fine-tuning specific solutions. On the other side of the spectrum, analytical approximations like the ones proposed by Shen can obtain solutions in computational times several orders of magnitude faster than the rest of the considered techniques, making then specially well-suited for the formulation of large-scale combinatorial problems. Nevertheless, they do not provide a feasible trajectory that complies with the dynamical model.

The RIC and SES strategies are used to numerically solve a set of test cases, which involve the minimization of the Δv spent during a transfer while complying with a maximum transfer time. Such tests cases are depicted in Table 9,

Table 9: Boundary conditions for the test cases

Case	Δt_{max} (days)	$\Delta\Omega_0$ (deg)	h_0 (km)	h_f (km)	i_0 (deg)	i_f (deg)
1	100	30	800	900	98	99
2	10	3	800	900	98	99
3	100	10	800	900	98	99
4	24.86	0.46	779.3	733.8	98.64	97.45

where h_0 and h_f are the initial and final orbital altitudes, respectively, and Δt_{max} is the maximum transfer time. Cases 1 and 2 have been extracted from Ref. [104], while Cases 3 and 4 have been ob-

tained from Ref. [94]. The values of the constants of the gravitational model used during the resolution of the test cases are: $J_2 = 1.082635854 \cdot 10^{-3}$, $\mu = 3.986004418 \cdot 10^5$ km^3/s^2, and $R_\oplus = 6378.137$ km. In addition, the considered value of the thrust acceleration for the propulsive arcs is $f = 3.5 \cdot 10^{-3}$ m/s^2.

Tables 10 and 11 show the Δv consumed by the RIC and SES strategies for the aforementioned test cases, as well as the solutions achieved by other methods provided in the literature.

Table 10: Method performance comparison (Test cases 1 and 2)

	Case 1	Case 2
Method	Δv (m/s)	Δv (m/s)
YSS [104]	596.7	507.0
Cerf [32]	598.1	652.5
SES [32]	598.2	664.3
RIC	598.8	567.2

Table 11: Method performance comparison (Test cases 3 and 4)

	Case 3	Case 4
Method	Δv (m/s)	Δv (m/s)
Shen (a,i) [94]	325.3	255.1
Shen (a,i,Ω) [94]	324.4	254.1
SES [32]	313.3	1187.6
RIC	313.5	245.0

The comparison of the RIC strategy with the other relevant methods can be summarized as follows:

1. SES: As anticipated in Section 3.3.2, the RIC strategy clearly outperforms the SES strategy for tight transfer times (Cases 2 and 4), while being marginally worse for longer transfer times (Cases 1 and 3). As both methods have a similar computational complexity, it is reasonable to prefer the RIC strategy.

2. Cerf: The Cerf approach improves the Δv achieved by the SES strategy, at the cost of a greater computational complexity. However, when compared to the RIC strategy, it shows the same disadvantages as the SES strategy on top of the increased computational complexity.

3. YSS: The YSS method outperforms the RIC strategy. This is a logical result because the YSS method is a strict generalization of the RIC, SES and Cerf methods. However, due to the computational complexity of its associated optimal control problem,

the YSS method might not be suitable for the formulation of large-scale combinatorial problems. Additionally, it has to be noted that several instances of the SES strategy are solved to obtain an initial guess of the costates of the YSS method. As the RIC strategy provides a Δv improvement over the SES strategy for longer transfer times, it might provide a better costate guess for those problem instances. Thus accelerating the resolution of the YSS method.

4. Shen: The RIC strategy outperforms both Shen analytical approximations. Therefore, choosing which of these methods to use in a large-scale combinatorial problem involves a trade-off between solution quality and computational efficiency.

The main objective of the proposed methodology is achieving an advantageous trade-off between computational efficiency and the optimality of the solution. Therefore, the next logical step is to figure out if the RIC strategy can successfuly use the heuristic control laws described in Section 3.2.3 to further accelerate the resolution of the method, while maintaining an advantageous Δv consumption. Tables 12, 13, 14, and 15 show the solutions of the four test cases, obtained by the SES strategy and the different control laws proposed for the RIC strategy. Specifically, the Δv consumed by each of those methods, as well as the Δa and Δi changes performed during the first propulsive arc are specified. Moreover, the depicted CPU times have been obtained with the SciPy Sequential Least Squares Programming solver in a PC featuring an Intel Core i7-1165G7 processor and 16 GB of RAM.

Table 12: Heuristic control laws performance comparison for Test case 1

Method	Δv (m/s)	Δa (km)	Δi (deg)	CPU time (s)
RIC (Optimal)	598.84	-391.77	1.216	0.103
RIC (Corr. Edelbaum)	599.05	-391.77	1.217	0.0383
RIC (Proportional)	600.06	-375.60	1.294	0.0445
SES [32]	598.23	-388.71	1.239	0.0125

Table 13: Heuristic control laws performance comparison for Test case 2

Method	Δv (m/s)	Δa (km)	Δi (deg)	CPU time (s)
RIC (Optimal)	567.23	-310.77	1.015	0.0464
RIC (Corr. Edelbaum)	567.25	-310.77	1.014	0.0185
RIC (Proportional)	567.78	-310.77	1.017	0.0297
SES [32]	664.28	-388.75	1.531	0.0148

Table 14: Heuristic control laws performance comparison for Test case 3

Method	Δv (m/s)	Δa (km)	Δi (deg)	CPU time (s)
RIC (Optimal)	313.46	-122.82	0.8957	0.0622
RIC (Corr. Edelbaum)	313.46	-122.82	0.8957	0.0410
RIC (Proportional)	313.46	-122.82	0.8957	0.0424
SES [32]	313.25	-121.20	0.9064	0.0183

Table 15: Heuristic control laws performance comparison for Test case 4

Method	Δv (m/s)	Δa (km)	Δi (deg)	CPU time (s)
RIC (Optimal)	245.03	-10.00	-0.3482	0.281
RIC (Corr. Edelbaum)	245.03	-10.00	-0.3482	0.0856
RIC (Proportional)	245.04	-9.98	-0.3482	0.0857
SES [32]	1187.6	-37.77	2.3004	0.382

It can be seen that both heuristic control laws of the RIC strategy achieve near-optimal solutions, with the corrected Edelbaum control law being marginally better, while achieving up to a threefold decrease in the CPU time with respect to the optimal control law. In turn, the SES approach is faster, except for the Test case 4. However, as all the shown computational times are suitable for the formulation of large-scale combinatorial problems and are roughly of the same order of magnitude, the potential Δv improvements of the RIC strategy make it reasonable to prefer it over the SES strategy.

It has to be noted that the difference in computational time between the heuristic and optimal implementations of the RIC strategy depends on the convergence trends of the algorithm described in Section 3.2.4. Specifically, this algorithm is able to consistently achieve the target semimajor axis (with an accuracy smaller than one meter) with a number of iterations of the root-finding algorithm between two and five. Moreover, three characteristic areas with diferent algorithmic behaviors have been observed as a function of the RAAN modified in the final propulsive arc.

First, when the RAAN variations are very small, the initial β angles achieve the same sign of ψ. Hence, the secant method is used and it is able to find the solution before a change in the sign of ψ is produced. This behavior has been observed in the resolution of Case 4.

Second, for larger RAAN variations, the initial β angles achieve different signs of ψ. Thus, Brent's method is directly used to find the solution. This behavior has been observed in the resolutions of Cases 1 to 3.

Third, for even larger RAAN variations, the initial β angles once again achieve the same sign of ψ and, just like in the first case, the secant method achieves the solution of the problem.

Finally, there are small transition zones between the aforementioned ones, in which the the initial β angles achieve the same sign of ψ but, after one of the iterations of the secant method, the sign of ψ changes, so the resolution is continued by Brent's method.

3.5 CONCLUSIONS

This manuscript proposes a novel J_2-perturbed continuous-thrust transfer between circular orbits. Specifically, this transfer considers tangential and out-of-plane thrust, with the thrust yaw angle being the sole control variable. Moreover, the dynamics of the transfer is described by two state variables, namely, the semimajor axis and the relative inclination with respect to the target orbit. The optimal solution of this problem involves finding a root of a function of the thrust yaw angle. Two heuristic control laws provide initial guesses for this root-finding problem. The convergence of this solution can be guaranteed with the use of bracketing root-finding methods, assuming that the function is continuous.

Furthermore, a thrust-coast-thrust strategy has been formulated. Its first propulsive arc corresponds to an Edelbaum transfer, while the second one performs the aforementioned relative inclination change transfer. This results in a three-variable NLP problem. The performance of this strategy for time-constrainted Δv-optimal transfers has been compared with the solutions of other relevant methods available in the literature. It has been shown that the proposed three-arc strategy provides an advantageous trade-off between solution quality and computational complexity, thus being suitable for the formulation of large-scale combinatorial problems. Finally, it has been shown that the use of the proposed heuristic control laws in the second propulsive arc results in near-optimal solutions, while further accelerating the resolution process.

Future work will evaluate the possibility of improving the fidelity of the proposed methodology by the consideration of effects such as Earth-shadow eclipses. Moreover, an analysis of the mathematical conditions, under which the direct modification of the RAAN is more advantageous than increasing the nodal precession, will be carried out.

Part III

PRELIMINARY MISSION ANALYSIS

I thought of a labyrinth of labyrinths, of one sinuous spreading labyrinth that would encompass the past and the future and in some way involve the stars.

— Jorge Luis Borges, The Garden of Forking Paths, 1962

4

LARGE-SCALE OBJECT SELECTION FOR DEBRIS REMOVAL

4.1 INTRODUCTION

Most objects in low Earth orbit are concentrated in certain privileged orbital regions. A high density of objects in these regions can result in collisions that generate new objects, thus increasing the possibility of subsequent collisions, and potentially leading to a cascade effect that can severely impact future space operations [65]. Further research has concluded that active removal of certain objects is necessary to achieve the stabilisation of the number of resident space objects in the most populated zones [71, 75].

Active removal missions will most likely not target a single object, but instead multiple objects could be deorbited in a single mission. This poses the question of which objects should be removed in a particular mission, and in which order. Likewise, it is also possible to study the joint optimisation of multiple deorbiting missions, each performed by a different satellite targeting either an initially fixed [31, 106] or undetermined [38] number of objects from a prescribed pool of candidate objects, or to minimise the number of deorbiting satellites required to remove all objects in the pool [99]. However, the main literature on the topic is focused on the optimisation of a single multi-objective mission, where the objects to be removed, the visiting sequence and the required manoeuvres must be determined.

Metaheuristic and heuristic methods, as well as branch-and-bound-based strategies have been proposed for the design of multi-objective space debris removal missions. Current approaches in the literature decide the objects to be removed prior to the sequence and manoeuvre optimisation [18, 77, 107, 108, 115], or consider a pool of candidate objects from which either an initially fixed [6, 28, 30, 43, 44, 83, 85, 95] or unknown [40, 58, 74, 76, 105] number of objects are selected. However, with only few exceptions [58, 74, 85], the size of the pool is unrealistically small in most of the aforementioned references, due to practical limitations of computational resources. Hence, the development of methods capable of dealing with a pool of objects that is large enough to be representative of the distribution of the most hazardous objects in low-Earth orbit (LEO), would provide a more realistic view of the actual deorbiting capabilities of multi-objective missions, and offer a valuable tool for the design of future active debris removal missions.

As already mentioned, the computational burden is the main impediment for increasing the size of the pool of candidate objects, as not only the complexity of the optimisation problem increases with the number of objects, but also reliably accounting for the non-linear orbital dynamics results in a non-negligible computational load. Consequently, orbital propagations predominantly rely upon analytical methods, which may account for the J_2 orbital perturbation [18, 28, 30, 40, 74, 85, 107], and only occasionally the atmospheric drag as well [40]. The J_2 perturbation is usually treated in averaged form, or linearised around reference solutions [30]. Further to this, for practical considerations, most bibliographical references construct the orbital transfers based solely on impulsive manoeuvres and coasting arcs. In an attempt to lower the computational cost, transfer options and their associated Δv are often pre-computed in a grid of prescribed epochs [40, 58, 76, 77, 85, 105]. Only in few cases low-thrust transfer trajectories have been considered, often in simplified form and with a small sized pool of objects [40, 50, 83, 115] in order to compensate for the additional computational burden; only recently Ref. [74] has managed to approach the problem with low-thrust transfers and a reasonably large pool of 597 objects.

In terms of the particular algorithms used to obtain the sequence of removed objects, the prevailing class of methods within the literature are metaheuristic algorithms in their various flavours, including: multi-objective particle swarm [38, 106, 107], metaheuristic inver-over [58], metaheuristic Physaurum algorithm [40], ant colony optimisation [95, 99], simulated annealing [31, 43] or genetic algorithms [44, 76, 83], to cite a few. Less often, heuristic methods have also been used, such as an heuristic series method [6], a greedy heuristic method [105] or a heuristic beam search [74]. According to the works covered in our literature review, only metaheuristic or heuristic strategies have been applied to solve problems with large pools of candidate objects. However, these strategies do not explore the whole search space of object combinations, whereas branch-and-bound techniques are able to explore or prune the whole search space to demonstrate the optimality of the selected object combination.

This manuscript proposes an innovative framework based on well-known Operational Research methodology that, given a large set of candidate objects with an associated threat value, selects a subset of these objects to be removed, and defines the trajectory that allows to rendezvous with them in an optimal order, so that the threat value of the removed objects is maximised, while a limit mission duration and a Δv budget are imposed as constraints. The proposed algorithm comprises two different levels: the upper level selects the objects to be removed so that their aggregated threat value (here characterised by their criticality index [91]) is maximised, whereas the lower level

checks the feasibility of time and Δv constraints while determining the mission sequence and trajectory.

On the one hand, the upper level is described by an Integer Programming problem, which selects the most promising subset of candidate objects. A novel formulation that avoids the appearance of solutions with subtours during the resolution of this problem has been devised. On the other hand, the lower level is described by a Mixed Integer Non-Linear Programming problem, which is broken down in an Integer Programming master problem and a Nonlinear Programming subproblem using Benders decomposition [17]. The master problem and the subproblem are iteratively solved to find tight upper and lower bounds to the optimal solution.

The efficiency of this algorithm lies in the fact that the no-subtour formulation allows the upper level to effectively select the most promising object subsets while the lower level can check their feasibility without having to reach the convergence of the upper and lower bounds of the Benders decomposition, which is the most challenging part of the problem because it encapsulates the nonlinearities of the orbital dynamics. As a result, we have obtained an efficient and scalable algorithm capable of dealing with a large-sized pool of up to several hundreds of candidate objects within an affordable computational time for the resolution of such large-scale problems, while performing exhaustive explorations of the search space of object combinations to demonstrate the optimality of the provided solution. Moreover, the modular structure of the proposed framework makes it possible to easily introduce additional features and improvements to the current process.

The remainder of this manuscript is organised as follows. Section 2 presents the detailed problem description and relevant concepts. Section 3 walks through the mathematical model, which is divided in the object selection model (Sec. 3.1) and the feasibility model (Sec. 3.2), and introduces the mathematical formulation and notation. In Section 4 the solution approach is presented, with separate subsections for the object selection (Sec. 4.1) and feasibility (Sec. 4.2) problems. Section 5 gathers the computational experiments and results, and Sec. 6 summarises the main conclusions of this work.

4.2 PROBLEM DESCRIPTION

Let us consider a pool of candidate spaceborne objects to be potentially removed from orbit. In brief, the problem consists on finding a subset of target objects within the pool, and the sequence in which these should be visited, as depicted in Figure 7, where each spot represents a candidate object and the arrows define the removal sequence of the selected objects. The solution of this problem seeks to maximise a given cost function while fulfilling certain constraints.

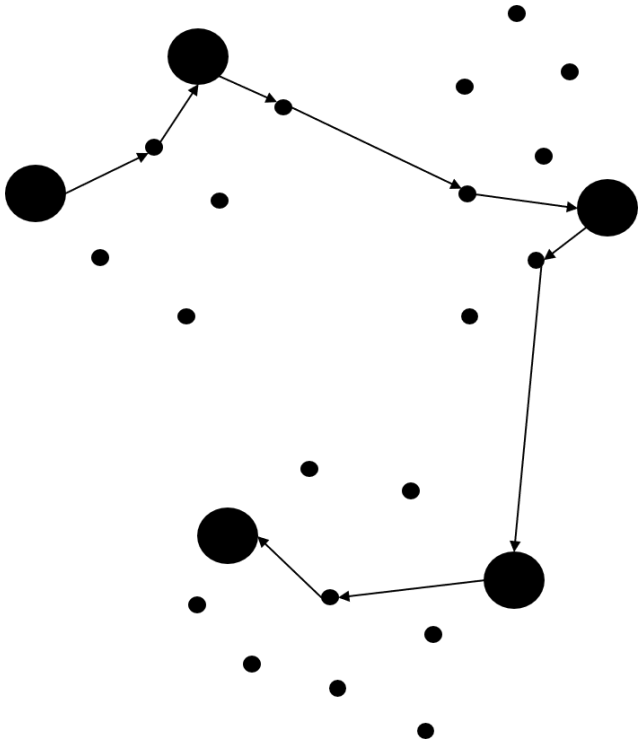


Figure 7: Multi-target debris removal problem.

Each object in the pool poses a different degree of threat and contributes differently to the risk of operations and sustainability of low-Earth orbit. This level of threat has to be quantified, so that a numerical criterion can be employed to decide the removal of which object offers a bigger reward in terms of an effective decrease of the threat level. Thus, each object is assigned a value, equivalent to a *criticality index*, which accounts for several factors such as their size, mass, lifetime, or how crowded the orbits where they are placed are. In Figure 7 the objects with a higher criticality index are depicted with thicker spots. The aggregated value of the criticality index of all removed objects provides the cost function to be maximised. Indeed, the removed debris has to be as threatening as possible so that the mission has an effective impact in the future evolution of the space environment [71, 75]. In the current work, the numerical value of such criticality index, or the criteria for their assignment, are not discussed and are assumed to be known.

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

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

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

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

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

As a consequence of that, the proposed problem maximises the reward obtained when removing the selected objects, which is a measurement of the threat that those objects pose to the space environment, while complying with a predefined Δv budget and a maximum mission duration. Then, this problem is divided in two parts,

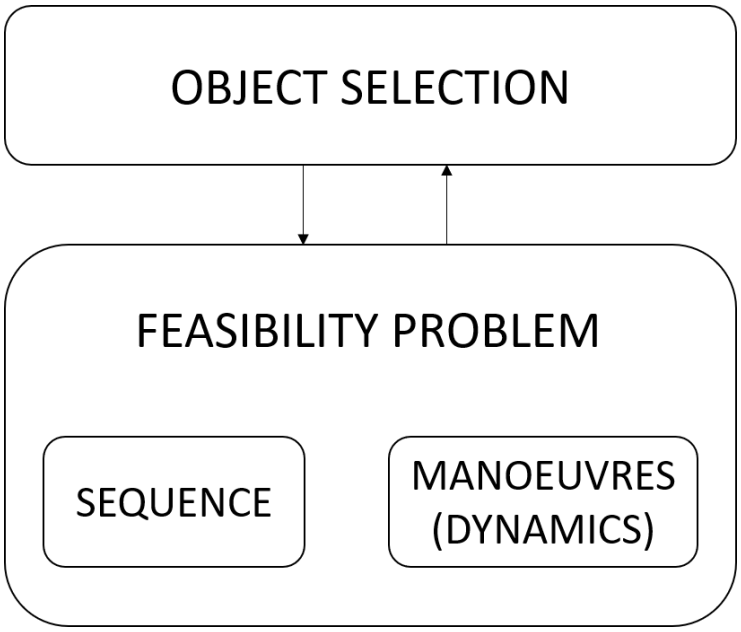


Figure 8: Problem structure.

as shown in Figure 8: 1) an *object selection* problem, which optimises the reward of the sequence of objects to be removed; and 2) a *feasibility* problem, which makes use of orbital mechanics to ensure that a mission profile, which complies with prescribed Δv budget and a mission duration constraints, can be generated with the selected objects to be removed. This way, the object selection problem passes promising sets of objects through to the feasibility problem, which accepts or rejects them, until the optimal sequence is found.

4.3 MATHEMATICAL MODEL

This section defines the mathematical formulation for the *object selection model* and the *feasibility model*, which correspond to the problems described above.

4.3.1 *Object Selection Model*

The aim of the *object selection model* is to select the optimal set of objects to remove, together with a potential removal sequence. The notation in the *object selection model* is defined as follows.

Sets

- D is the set of candidate objects to be visited indexed by i or j.

Parameters

- r_i is the reward obtained when removing object $i \in D$.

- Δv_{ij} is an estimation of the cost (i.e., Δv) of performing a transfer from object $i \in D$ to object $j \in D$.
- Δv_T is the Δv budget (i.e., the available amount) for the whole mission.

Variables

- $y_i \in \{0, 1\}$ is 1 if object $i \in D$ is removed; 0, otherwise.
- $x_{ij} \in \{0, 1\}$ is 1 if a transfer between objects $i \in D$ and $j \in D$ is performed; 0, otherwise.

The aim of the model is to maximise the total reward corresponding to the visited objects in the mission, i.e, the sum of the individual rewards of each of the objects. Hence, the objective function is as shown in (80).

$$\max \left\{ \sum_{i \in D} r_i \, y_i \right\} \tag{80}$$

Constraints (81) ensure that the estimated spent total Δv during the mission does not exceed the available budget, Δv_T.

$$\sum_{i \in D} \sum_{\substack{j \in D \\ i \neq j}} \Delta v_{ij} \, x_{ij} \leqslant \Delta v_T \tag{81}$$

Constraints (82) state that a transfer between two objects cannot be performed in both directions in the same mission.

$$x_{ij} + x_{ji} \leqslant 1 \quad \forall i \in D, \forall j \in D : (i \neq j) \tag{82}$$

Constraints (83) ensure that a transfer has to connect two different objects, i.e., it cannot connect an object with itself.

$$x_{ii} = 0 \quad \forall i \in D \tag{83}$$

Constraints (84) say that if an object is visited, it must be visited with a transfer from another object. Note that, to initialise the mission, a dummy object ($i = 1$) is employed; it is located in the parking orbit of the satellite and set to be the first one in the sequence (i.e., $y_1 = 1$, $\sum_{j \in D : j > 1} x_{1j} = 1$ and $x_{i1} = 0 \quad \forall i \in D$); it is removed without spending any Δv and without providing any reward. Similarly, Constraints (85) state that every visited object must be followed by another object unless it is the last one in the mission (i.e., there is no outgoing transfer for the last object in the sequence).

$$\sum_{\substack{j \in D \\ i \neq j}} x_{ji} = y_i \quad \forall i \in D : (i > 1) \tag{84}$$

$$\sum_{\substack{j\in D\\ i\neq j}} x_{ij} \leqslant y_i \quad \forall i \in D : (i > 1) \tag{85}$$

As a final remark, it has to be noted that this mathematical model can be easily extended to include additional features. For instance, compliance of the servicing satellite with end-of-life guidelines can be enforced; this can be done by the simple expedient of imposing the removal of a final virtual object a (i.e., $y_a = 1$) for which there is no outgoing transfer (i.e., $x_{ai} = 0, \quad \forall i \in N$), and such that it represents the desired final orbit.

Another plausible addition is the consideration of a fixed Δv cost associated to proximity operations; this can be easily implemented by replacing Constraint (81) with the more general Constraint (86), which includes, for every removed object, a penalty Δv_j^{po} to account for the extra propellant consumed during co-orbiting, i.e.

$$\sum_{i\in D} \sum_{\substack{j\in D\\ i\neq j}} (\Delta v_{ij} + \Delta v_j^{po})\, x_{ij} \leqslant \Delta v_T \tag{86}$$

4.3.2 *Feasibility model*

The aim of the *feasibility model* is to check the attainability of the selected set of objects with the potential removal sequence or any variation of it.

As mentioned before, the mission profile comprises two kinds of stages for each object to be removed, namely, co-orbiting arcs and transfer arcs. The co-orbiting arc describes the satellite's trajectory while co-orbiting the object to be removed. The transfer arc describes the outgoing trajectory, which begins when the satellite leaves the orbit of the removed object and ends when the next object's orbit is intercepted. Impulsive manoeuvres are performed in the intersection point of each pair of arcs. Note that the last object to be removed does not have a transfer arc. This conceptual mission profile is depicted in Figure 9.

The notation in the *feasibility model* is defined as follows.

Sets

- $D' \subset D$ is a set composed of the objects to be removed, which are provided by the object selection problem. It is indexed by d.
- L is a set such that $|L| = |D'|$. It is employed to store the order in which the objects in D' will be removed. Note that the potential removal sequence provided by the *object selection model* is not necessarily maintained, i.e., it might be updated in the *feasibility model*. It is indexed by ℓ.

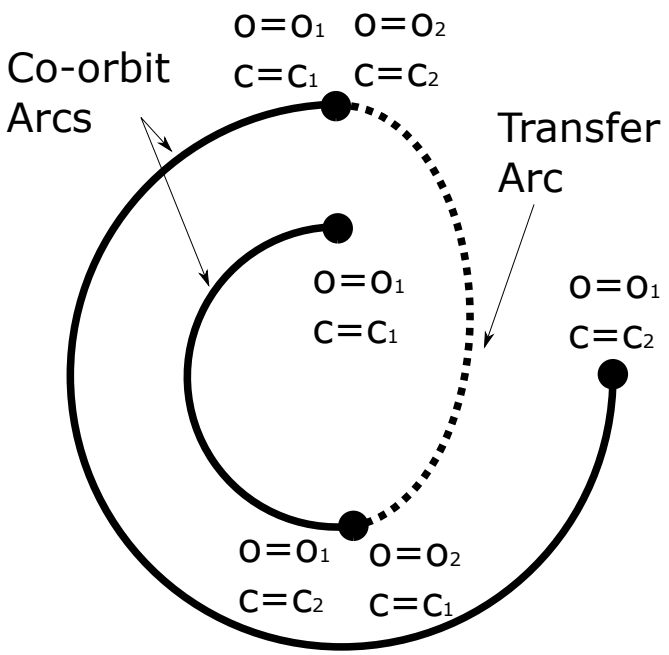


Figure 9: Mission profile nomenclature.

- $O = \{o_1, o_2\}$ is the set of arc types. The first element in the set stands for co-orbit arcs and the second one for transfer arcs. It is indexed by o.
- $C = \{c_1, c_2\}$ is the set that identifies the intersection or crossing points between arcs, that is, the points where impulsive manoeuvres are performed. The first element in the set stands for the initial point of the arc, and the second one for its final point. It is indexed by c.

Parameters

- μ is the gravitational constant of the Earth.
- $R_\oplus$ is the equatorial radius of the Earth.
- J_2 is the spherical harmonic of degree 2 and order 0 of the Earth's gravity field.
- Δt is the minimum co-orbiting time, i.e. the minimum time lapse during which the satellite has to remain in close operations with an object to successfully eliminate it.
- t_{max} is the maximum mission duration.
- t_d^0 is the reference time in which the orbital parameters of the object $d \in D'$ are known.

- a_d^0, e_d^0, i_d^0, Ω_d^0, ω_d^0 and M_d^0 are the classical orbital elements of the object $d \in D'$ at time t_d^0, where M_d^0 stands for the mean anomaly and the other elements have their usual meaning (i.e., semi-major axis, eccentricity, inclination, right ascension of ascending node and argument of perigee, respectively).

Variables

- $s_{d\ell} \in \{0, 1\}$ is 1 if the object $d \in D'$ is visited in position $\ell \in L$ of the sequence; 0, otherwise.
- $a_{\ell o}$, $e_{\ell o}$, $i_{\ell o}$, $n_{\ell o}$ and $p_{\ell o}$ are, respectively, the semi-major axis, eccentricity, inclination, mean motion and semilatus rectum of the orbit $o \in O$ associated to the object visited in position $\ell \in L$.
- $\Omega_{\ell oc}$, $\omega_{\ell oc}$ and $M_{\ell oc}$ are, respectively the right ascension of the ascending node, argument of perigee and mean anomaly of the point $c \in C$ in the orbit $o \in O$ associated to the object visited in position $\ell \in L$.
- t_ℓ^{ref} is the reference time in which the osculating orbital elements of the object visited in position $\ell \in L$ are known.
- $t_{\ell oc}$ is the time in which the satellite is at the point $c \in C$ in the orbit $o \in O$ associated to the object visited in position $\ell \in L$.
- $\vec{r}_{\ell oc}$ is the position vector of the point $c \in C$ in the orbit $o \in O$ associated to the object visited in position $\ell \in L$.
- $\vec{v}_{\ell oc}$ is the velocity vector of the satellite when it is at the point $c \in C$ in the orbit $o \in O$ associated to the object visited in position $\ell \in L$.
- Δv_ℓ^{tran} is the magnitude of the $\Delta\vec{v}$ vector of the manoeuvre performed to depart from the orbit where the satellite is co-orbiting the object visited in position $\ell \in L$.
- Δv_ℓ^{ren} is the magnitude of the $\Delta\vec{v}$ vector of the manoeuvre performed when intercepting the object visited in position $(\ell + 1) \in L$ in order to start co-orbiting it.
- z is the optimal value of the objective function of the feasibility problem.

The *feasibility model* minimises the total Δv consumed throughout the mission as shown in objective function (87).

$$\min z = \sum_{\substack{\ell \in L \\ \ell < |L|}} \Delta v_\ell^{ren} + \sum_{\substack{\ell \in L \\ \ell < |L|}} \Delta v_\ell^{tran} \tag{87}$$

Constraints (88) establish that for each position in the sequence there must be an object to be removed. Therefore, each object within D'

matches exactly one object within L. The orbit the satellite initially describes is modelled as a dummy object $d = 1$, which is automatically removed at the initial time. As a result, s_{11} is fixed to 1. Similarly, orbital time is fixed for the dummy object as follows: $t_{1o_1c_1} = t_1^{ref}$ and $t_{1o_1c_1} \leqslant t_{1o_1c_2}$.

$$\sum_{d \in D'} s_{d\ell} = 1 \qquad \forall \ell \in L : \ell > 1 \tag{88}$$

Variables $a_{\ell o}$, $e_{\ell o}$, and $i_{\ell o}$ remain constant during each of the orbital arcs; these variables for the co-orbiting arcs, as well as the reference time t_ℓ^{ref} necessary to propagate the rest of orbital parameters, are calculated in Equations (89).

$$\begin{bmatrix} a_{\ell o_1} \\ e_{\ell o_1} \\ i_{\ell o_1} \\ t_\ell^{ref} \end{bmatrix} = \sum_{d \in D'} s_{d\ell} \begin{bmatrix} a_d^0 \\ e_d^0 \\ i_d^0 \\ t_d^0 \end{bmatrix} \qquad \forall \ell \in L \tag{89}$$

Equations (90) und (91) compute the mean motion and the semilatus rectum for each object visited in position ℓ and orbit, respectively.

$$n_{\ell o} = \sqrt{\frac{\mu}{a_{\ell o}^3}} \qquad \forall \ell \in L, \forall o \in O : ((\ell < |L|) \vee (o < |O|)) \tag{90}$$

$$p_{\ell o} = a_{\ell o}\left(1 - e_{\ell o}^2\right) \quad \forall \ell \in L, \forall o \in O : ((\ell < |L|) \vee (o < |O|)) \tag{91}$$

Equations (92) calculate, for the co-orbiting arcs and for each object visited in position ℓ and intersection point, $\Omega_{\ell o_1 c}$, $\omega_{\ell o_1 c}$ and $M_{\ell o_1 c}$ for any time $t_{\ell o_1 c}$. That is, they perform a propagation of the state at reference time t_ℓ^{ref}, determined by $s_{d\ell}$ variables, to the state at $t_{\ell o_1 c}$.

$$\begin{bmatrix} \Omega_{\ell o_1 c} \\ \omega_{\ell o_1 c} \\ M_{\ell o_1 c} \end{bmatrix} = \sum_{d \in D'} s_{d\ell} \begin{bmatrix} \Omega_d^0 \\ \omega_d^0 \\ M_d^0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ n_{\ell o_1}\left(t_{\ell o_1 c} - t_\ell^{ref}\right) \end{bmatrix} + \\ + \frac{3 n_{\ell o_1} R_\oplus^2 J_2}{4 p_{\ell o_1}^2}\left(t_{\ell o_1 c} - t_\ell^{ref}\right) \begin{bmatrix} -2\cos\left(i_{\ell o_1}\right) \\ 4 - 5\sin^2\left(i_{\ell o_1}\right) \\ \sqrt{1 - e_{\ell o_1}^2}\left(3\sin^2\left(i_{\ell o_1}\right) - 2\right) \end{bmatrix} \tag{92}$$

$$\forall \ell \in L, \forall c \in C$$

Constraints (93) und (94) impose that position must be conserved before and after each manoeuvre, i.e., in the intersection points of the different trajectory arcs. Similarly to the position conservation, time between consecutive co-orbit and transfer arcs must be conserved, as stated in Constraints (95) und (96), respectively.

$$\vec{r}_{\ell o_1 c_2} = \vec{r}_{\ell o_2 c_1} \qquad \forall \ell \in L : (\ell < |L|) \tag{93}$$

$$\vec{r}_{\ell o_2 c_2} = \vec{r}_{(\ell+1) o_1 c_1} \qquad \forall \ell \in L : (\ell < |L|) \tag{94}$$

$$t_{\ell o_1 c_2} = t_{\ell o_2 c_1} \qquad \forall \ell \in L : (\ell < |L|) \tag{95}$$

$$t_{\ell o_2 c_2} = t_{(\ell+1) o_1 c_1} \qquad \forall \ell \in L : (\ell < |L|) \tag{96}$$

In order to impose the feasibility for transfer arcs between co-orbit arcs, Equations (97) establish mathematical conditions, by means of function $\vec{F}$, between Keplerian orbital elements and Cartesian position and velocity vectors at each orbit and intersection point for each object visited in position ℓ. Specifically, the function $\vec{F}$ is the composition of three operations, namely: 1) solving Kepler's equation to relate mean and eccentric anomalies [14, sec 4.3]; 2) projecting the Cartesian position into perifocal frame coordinates [14, sec 4.3]; and 3) transforming from perifocal to (equatorial) inertial frame coordinates [14, sec 2.1].

$$\begin{aligned} \begin{bmatrix} \vec{r}_{\ell oc} \\ \vec{v}_{\ell oc} \end{bmatrix} &= \vec{F}\left(a_{\ell o}, e_{\ell o}, \Omega_{\ell oc}, \omega_{\ell oc}, M_{\ell oc}\right) \\ \forall \ell \in L, \forall o \in O, \forall c \in C : ((\ell < |L|) \vee (o < |O|)) & \end{aligned} \tag{97}$$

Then, Equations (98) define the analytical propagation of Keplerian orbital elements for transfer arcs.

$$\begin{aligned} \begin{bmatrix} \Omega_{\ell o_2 c_2} \\ \omega_{\ell o_2 c_2} \\ M_{\ell o_2 c_2} \end{bmatrix} = \begin{bmatrix} \Omega_{\ell o_2 c_1} \\ \omega_{\ell o_2 c_1} \\ M_{\ell o_2 c_1} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ n_{\ell o_2}\left(t_{\ell o_2 c_2} - t_{\ell o_2 c_1}\right) \end{bmatrix} + \\ + \frac{3 n_{\ell o_2} R_{\oplus}^2 J_2}{4 p_{\ell o_2}^2}\left(t_{\ell o_2 c_2} - t_{\ell o_2 c_1}\right) \begin{bmatrix} -2\cos\left(i_{\ell o_2}\right) \\ 4 - 5\sin^2\left(i_{\ell o_2}\right) \\ \sqrt{1 - e_{\ell o_2}^2}\left(3\sin^2\left(i_{\ell o_2}\right) - 2\right) \end{bmatrix} \\ \forall \ell \in L : (\ell < |L|) \end{aligned} \tag{98}$$

The individual Δv spent in each of the manoeuvres is computed with Equations (99) und (100).

$$\Delta v_{\ell}^{ren} = \left|\vec{v}_{(\ell+1) o_1 c_1} - \vec{v}_{\ell o_2 c_2}\right| \qquad \forall \ell \in L : (\ell < |L|) \tag{99}$$

$$\Delta v_{\ell}^{tran} = \left|\vec{v}_{\ell o_2 c_1} - \vec{v}_{\ell o_1 c_2}\right| \qquad \forall \ell \in L : (\ell < |L|) \tag{100}$$

Note that the satellite must co-orbit each object in close operation for a minimum amount of time, Δt, in order to be able to deorbit it. Constraints (101) ensure this requirement is fulfilled. And Constraints (102) ensure that the transfer arc finishes after it has started.

$$t_{\ell o_1 c_1} + \Delta t \leqslant t_{\ell o_1 c_2} \qquad \forall \ell \in L \tag{101}$$

$$t_{\ell o_2 c_1} \leqslant t_{\ell o_2 c_2} \qquad \forall \ell \in L : (\ell < |L|) \tag{102}$$

Finally, the ending time of the mission must be lower than the maximum allowable mission duration, as stated in Constraint (103).

$$t_{|L| o_1 c_2} \leqslant t_{max} \tag{103}$$

4.4 SOLUTION APPROACH

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

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

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

4.4.1 *Object selection: Subtour management*

Constraints (84) und (85) of the object selection problem impose that, if an object is removed, it has at most one incoming and one outgoing transfer arc. However, this is not enough to correctly model the proposed mission because closed loops disjoint from the main rectilinear path of transfers would also fulfil that condition. Such disjoint loops are usually referred as *subtours* and constitute a very serious difficulty for the problem at hand. Figure 10 illustrates this concept. Representing the Δv_{ij} parameter as the distance between spots, the black arrows and dashed lines form a feasible rectilinear path of transfers. However, the grey arrows are shorter (require less Δv_{ij}) than the dashed lines, and as a result this solution with subtours would always be preferred over the presented rectilinear path.

The reason why the appearance of subtours is so negative is that plane changes are the most Δv intensive manoeuvres and it is expected that objects that provide important rewards r_i are spread across different orbital planes. As a result, the best solutions of this model are potentially the ones that form subtours involving objects within similar orbital planes; indeed, solutions tend to favour such configurations as expensive change-of-plane manoeuvres are thus avoided and replaced by cheaper manoeuvres across nearly coplanar planes. This phenomenon produces a great number of invalid solutions that have to be pruned out before achieving the best solution without subtours, which requires a great computational effort.

Two alternatives are proposed to manage this problem. On the one hand, the usual method to deal with subtours is considered in the upcoming subsection. On the other hand, a new linear formulation for

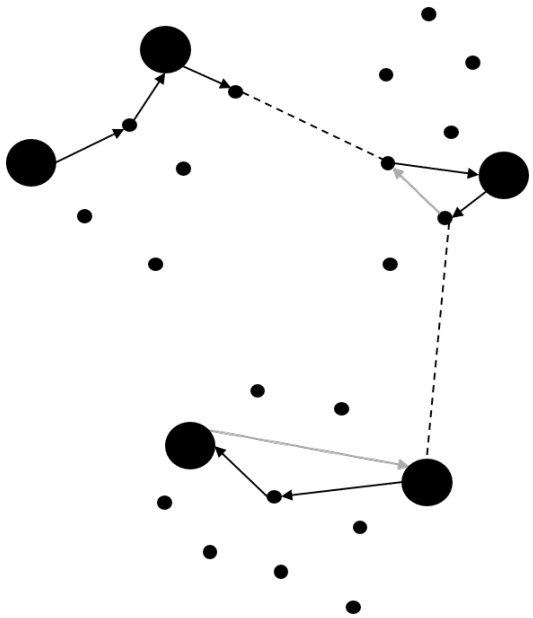


Figure 10: Diagram illustrating the appearance of subtours.

the object selection problem is presented in Sec. 4.1.2, which avoids the appearance of subtours.

4.4.1.1 *Classical subtour elimination*

Current Mixed Integer Linear Programming (MILP) formulations of this kind of models cannot prevent the natural appearance of subtours. The usual approach to deal with them is that, once a feasible solution is found, it gets checked whether it includes any subtours. If it does not, the solution is accepted; if subtours are found, the solution is rejected and a new constraint (104) is added to the problem for each of the subtours found.

$$\sum_{i \in R} \sum_{j \in R} x_{ij} \leqslant |R| - 1 \tag{104}$$

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

4.4.1.2 *No-subtour formulation*

The use of no-subtour Mixed Integer Linear Programming formulations has proven useful in Travelling Salesman Problems [80] as well as Time-Dependent Travelling Salesman Problems [48]. As a result,

Algorithm 1 Classical subtour elimination algorithm

```
1:  while solving Object selection problem do
2:      if feasible solution is found then
3:          Check subtours
4:          if Subtours are found then
5:              Reject solution
6:              Add subtour elimination constraints
7:          else
8:              Accept solution
9:          end if
10:     end if
11: end while
```

a new formulation has been devised that avoids the natural appearance of subtours for problems that only require visiting a subset of the candidate objects (unlike in Travelling Salesman Problems, where all candidate objects must be visited). It makes use of a new set $k \in K$ that orders the transfers from '1' to a maximum number of transfers to be made, $|K|$. Then, the discrete variables that represent each of the transfers x_{ij} are changed by new discrete variables x_{ijk} that are '1' when a transfer is made from object $i \in D$ to $j \in D$ in ordinal position $k \in K$, or '0' otherwise. It has to be noted that $\sum_{k \in K} x_{ijk}$ is equivalent to previous variables x_{ij}. This change is introduced in Constraints (81) bis (85).

In addition, it needs to be imposed that for each position $k \in K$ there is at most one transfer, as shown in Constraints (105).

$$\sum_{i \in D} \sum_{\substack{j \in D \\ i \neq j}} x_{ijk} \leqslant 1 \qquad \forall k \in K \tag{105}$$

The appearance of subtours is prevented by means of adding Constraints (106). They impose that, if there is a transfer in position $(k+1) \in K$, its initial object is the last of the transfers in position $k \in K$. In turn, if there is not a transfer in position $k \in K$, then there is no transfer in $(k+1) \in K$ either.

$$\sum_{\substack{i \in D \\ i \neq j}} x_{ijk} \geqslant \sum_{\substack{i \in D \\ i \neq j}} x_{ji(k+1)} \qquad \forall j \in D : (j > 1), \quad \forall k \in K : (k < |K|) \tag{106}$$

4.4.2 *Feasibility problem: Benders decomposition*

The feasibility problem is a Mixed Integer Nonlinear Programming problem (MINLP), i.e., it is formulated with nonlinear objective function and constraints while including continuous and discrete variables. Currently, this kind of problems is intrinsically difficult to solve. We propose a Benders decomposition based approach to overcome the difficulty.

Benders decomposition [17, 52] is a technique in mathematical programming that allows the solution of large and difficult problems that have a special block structure. It decomposes the problem in two different subproblems that are easier to solve. This method solves iteratively these two problems in order to achieve the optimal solution of the original problem.

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

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

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

4.4.2.1 *Subproblem*

The subproblem is composed of the objective function in (87) and Constraints (88) to (103). Note that, binary variables $s_{d\ell}$, which are

allocated to the master problem, are substituted by continuous variables $s^{rel}_{d\ell}$, as shown in Equations (107) und (108).

$$\begin{bmatrix} a_{\ell o_1} \\ e_{\ell o_1} \\ i_{\ell o_1} \\ t^{ref}_{\ell} \end{bmatrix} = \sum_{d \in D'} \left(s^{rel}_{d\ell} \begin{bmatrix} a^0_d \\ e^0_d \\ i^0_d \\ t^0_d \end{bmatrix} \right) \qquad \forall \ell \in L \tag{107}$$

$$\begin{bmatrix} \Omega_{\ell o_1 c} \\ \omega_{\ell o_1 c} \\ M_{\ell o_1 c} \end{bmatrix} = \sum_{d \in D'} \left(s^{rel}_{d\ell} \begin{bmatrix} \Omega^0_d \\ \omega^0_d \\ M^0_d \end{bmatrix} \right) + \begin{bmatrix} 0 \\ 0 \\ n_{\ell o_1} \left(t_{\ell o_1 c} - t^{ref}_{\ell} \right) \end{bmatrix} + + \frac{3 n_{\ell o_1} R^2_{\oplus} J_2}{4 p^2_{\ell o_1}} \left(t_{\ell o_1 c} - t^{ref}_{\ell} \right) \begin{bmatrix} -2 \cos\left(i_{\ell o_1}\right) \\ 4 - 5 \sin^2\left(i_{\ell o_1}\right) \\ \sqrt{1 - \left(e_{\ell o_1}\right)^2} \left(3 \sin^2\left(i_{\ell o_1}\right) - 2\right) \end{bmatrix}$$

$$\forall \ell \in L, \forall c \in C \tag{108}$$

Moreover, Constraints (109) are introduced in the subproblem. Their purpose is to fix the value of the $s^{rel}_{d\ell}$ variables with the values obtained in the master problem, namely $\hat{s}_{d\ell}$. The variables $\lambda_{d\ell}$, depicted in parenthesis next to constraints (109), are the dual variables of such constraints (i.e., its Lagrange multipliers). Their value is obtained when the subproblem is solved and their purpose is to provide information about the variation of the value of the objective function (87) for different object sequences. This information is employed to define the Benders optimality cuts in the master problem.

$$s^{rel}_{d\ell} = \hat{s}_{d\ell} \qquad (\lambda_{d\ell}) \qquad \forall d \in D', \quad \forall \ell \in L \tag{109}$$

Note that the subproblem cannot be infeasible because there are not limitations on the Δv to be used. This limitation is only imposed in the *object selection* model and checked in the feasibility problem once the solution is known or using the information provided by the lower and upper bounds of the algorithm as shown in Algorithm 2.

4.4.2.2 *Master problem*

The master problem is composed of objective function (110), constraints (88) and Benders optimality cuts (111), where z_{MM} represents its optimal value.

$$\min z_{MM} = \theta \tag{110}$$

The objective function (110) only comprises a dummy variable θ, whose value is driven by Benders optimality cuts (111), which in fact are an approximation of the subproblem. Note that these cuts are dynamically generated making use of the information provided by the

subproblem, by means of the $\lambda_{d\ell}$ variables, in order to choose the next object sequence to be evaluated. The following additional notation is necessary to define these cuts:

- $\hat{s}^b_{d\ell}$: optimal value of $s_{d\ell}$ variables obtained in the master problem at iteration $b \in B$.
- $\hat{z}^b$: value of the subproblem objective function z at iteration $b \in B$.
- $\hat{\lambda}^b_{d\ell}$: optimal value of dual variables $\lambda_{d\ell}$ in the subproblem at iteration $b \in B$.

$$\theta \geqslant \hat{z}^b + \sum_{d \in D'} \sum_{\ell \in L} \left(s_{d\ell} - \hat{s}^b_{d\ell}\right) \hat{\lambda}^b_{d\ell} \qquad \forall b \in B \tag{111}$$

4.4.2.3 *Feasibility algorithm*

Algorithm 2 shows the proposed iterative approach for solving the feasibility problem. For a given set of objects to be deorbited, it iteratively solves the master problem and the subproblem until the proposed set of objects is either rejected because it is infeasible due to dynamics or accepted. It has to be noted that the number of iterations b cannot be greater than $|B|$, this being a conservative upper bound. This is a programming convention that prevents the possible appearance of infinite loops and facilitates the detection of implementation errors; nonetheless, for a correct implementation the algorithm should typically finish long before $|B|$ iterations are reached.

4.4.3 *Complete algorithms*

To sum up, this section presents the algorithms that consolidate the whole framework. Algorithm 3 integrates the resolution of the Feasibility Problem, shown in Algorithm 2, with the subtour elimination strategy displayed in Algorithm 1. In turn, Algorithm 4 represents the whole resolution process when the no-subtour formulation is used.

4.5 COMPUTATIONAL EXPERIMENTS AND RESULTS

This section intends to demonstrate the performance of the methodology developed in the preceding sections. First, the set-up of the proposed numerical tests is presented and described; this includes a test case with a pool of 1000 candidates objects (Test Case 1) aimed at demonstrating the capability of solving problems with large pool sizes, and another scenario (Test Case 2) intended to provide a more realistic problem set-up based on the actual low-Earth orbit environment; additionally, the estimation of Δv_{ij} is addressed in necessary detail. Second, the performance of the developed algorithm

Algorithm 2 Feasibility problem algorithm

$LB^0 = -\infty$
$UB^0 = \infty$
for $b = 1$ to $|B|$, $b+$ **do**
    Solve Master Problem **Obtaining** $s_{d\ell}, z_{MM}$
    $\widehat{s}_{d\ell} = s_{d\ell}$
    $LB^b = \max\{LB^{b-1}, z_{MM}\}$
    if $LB^b \geqslant \Delta v_T$ **then**
        Reject solution
        Stop
    end if
    Solve Subproblem **Obtaining** z
    $UB^b = \min\{UB^{b-1}, z\}$
    if $UB^b \leqslant \Delta v_T$ **then**
        Accept solution
        Stop
    end if
    $\widehat{s}^b_{d\ell} = s_{d\ell}$
    $\widehat{\lambda}^b_{d\ell} = \lambda_{d\ell}$
    $\widehat{z}^b = z$
    Generate Benders cut
end for

Algorithm 3 Complete algorithm for subtour elimination strategy

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

Algorithm 4 Complete algorithm for no-subtour formulation

while solving Object selection problem **do**
    if feasible solution is found **then**
        Solve Feasibility Problem
    end if
end while

is assessed, first by analysing its behaviour for the Object Selection Problem alone (i.e. excluding the Feasibility Problem, so that the no-subtour and subtour elimination formulations can be compared), and later by employing the full algorithm, applied to the two aforementioned test cases; these tests allow to draw relevant conclusions on the performance and suitability of the algorithm for the intended tasks. The algorithm has been implemented in GAMS 25.1.1 and solved with CONOPT 4.05 and CPLEX 12.8.0 in a PC featuring an Intel Core i5-7500 and 8GB of RAM. Computations have been performed using a single thread with a maximum CPU frequency of 3.4 GHz.

4.5.1 *Set-up of Numerical Tests*

4.5.1.1 *Test Case 1: Performance Assessment*

The aim of this test case is to demonstrate the capability of our algorithm for solving problems with large object pool sizes. For this purpose, this test case considers 1000 candidate objects and a maximum number of 10 objects to be removed. The reason for this limitation is to give preference to removing the 10 most impactful objects that available resources allow, over removing a greater number of less impactful objects with alike orbital characteristics; indeed, the latter could result in a diminishing returns phenomenon, where eliminating a larger number of objects would not be as effective, in terms of the aggregated removed threat based on the summed criticality of the removed objects.

In brief, this mission scenario is defined by the following parameters:

$$|D| = 1000, \qquad |K| = 10, \qquad t_{max} = 1 \text{ year}, \qquad \Delta t = 1 \text{ hour}$$

The reference orbital parameters associated to each of the 1000 candidate objects correspond to circular orbits in LEO. Their orbital parameters, as well as the rewards obtained when removing these objects, are randomly generated following a uniform distribution with the following parameters[1]:

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

Purely Keplerian dynamics is used for this test case. In terms of computational cost, the qualitative effect of considering a J_2-perturbed

1 For the sake of reproducibility, this dataset is made available online along with this article.

dynamics versus a purely Keplerian motion is twofold: On the one hand, nodal precession makes it possible to achieve object combinations that would be infeasible in the purely Keplerian case, thus accelerating the pruning of the search space. On the other hand, the Δv_{ij} estimation for a J_2-perturbed problem would be less accurate, thus passing more infeasible object combinations through to the feasibility problem. However, in practice none of these effects has any meaningful nor detrimental influence in terms of the qualitative performance of the algorithm, since the combinatorial complexity of the problem remains the same both for the Keplerian and J_2-perturbed cases.

As a consequence, the J_2-perturbed and Keplerian problems provide a similar insight on the algorithm performance. However, the feasibility problem is very nonlinear and nonconvex, thus making it highly dependent on a good initial guess. Since good initial guesses for Keplerian dynamics are easier to obtain, the use of Keplerian dynamics helps to uncouple the selection of Δv_{ij} estimations and initial guesses from the assessment of the performance of the proposed algorithm. Essentially, this results in a more reliable measurement of the performance of the framework when dealing with a very large combinatorial complexity, and thus justifies the decision of using Keplerian motion for the scope of these numerical tests.

4.5.1.2 *Test Case 2: Realistic Mission Application*

This test case is intended to assess the performance of the proposed methodology in a scenario with a more realistic distribution of objects. This case also considers Keplerian dynamics and a maximum of 10 objects to be removed; however, the pool is built by filtering objects from the SATCAT database according to the following criteria:

- Earth orbiting objects.
- Operational status “-" (nonoperational) or “ " (unknown).
- Apogee altitude less than 2000 km.
- Perigee altitude greater than 200 km.
- Radar Cross Section (RCS) greater than 1 m^2.

This results in a pool of 1571 objects. The reward r_i associated to each of them is computed following the Criticality of Spacecraft Index described in [91]. In order to compute this index, it is necessary to know the altitude, inclination and mass of each object; however, the SATCAT database does not provide mass information, so the SATCAT RCS data has been used as a proxy for the object mass. After computing the criticality index for the previous set of objects, only

those with a criticality index greater than 0.1 were selected to configure the application case, resulting in a reduced dataset of 420 objects[2]. Orbital parameters of those objects are obtained from Two Line Elements (TLEs).

4.5.1.3 *Estimation of* Δv_{ij}

The object selection problem requires an estimation of the Δv_{ij} consumed in each transfer. Since the feasibility problem considers two-impulse transfers and the proposed data sets comprise circular or nearly circular orbits, these estimations have been made assuming Hohmann transfers with optimal split plane changes. This concept is a generalisation of Hohmann transfers in which the initial and final orbits are not coplanar. The velocities of the initial and final orbits, namely:

$$v_i = \sqrt{\frac{\mu}{R_i}}, \qquad v_j = \sqrt{\frac{\mu}{R_j}} \tag{112}$$

and the velocities at apogee and perigee of the transfer orbit, i.e.:

$$v'_i = \sqrt{\frac{2\mu}{R_i}\frac{R_j}{R_i + R_j}}, \qquad v'_j = \sqrt{\frac{2\mu}{R_j}\frac{R_i}{R_i + R_j}} \tag{113}$$

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

$$\Delta v_{ij} = \sqrt{v_i^2 + v_i'^2 - 2v_i v'_i \cos(\phi)} + \sqrt{v_j^2 + v_j'^2 - 2v_j v'_j \cos(\theta_p - \phi)} \tag{114}$$

where θ_p is the angle between the initial and final orbital planes, and ϕ is the change of plane performed during the first impulsive manoeuvre. The optimal value for ϕ lies at a stationary point, $d\left(\Delta v_{ij}\right)/d\phi = 0$, which results in the following equation, that typically needs to be solved numerically:

$$\frac{v_i v'_i \sin(\phi)}{\sqrt{v_i^2 + v_i'^2 - 2v_i v'_i \cos(\phi)}} - \frac{v_j v'_j \sin(\theta_p - \phi)}{\sqrt{v_j^2 + v_j'^2 - 2v_j v'_j \cos(\theta_p - \phi)}} = 0 \tag{115}$$

It can be observed that, if $\phi = 0$, the first term of left hand side of Eq. (115) vanishes and the value of the second term becomes negative. In turn, if $\phi = \theta_p$, the second term vanishes and the first term becomes positive. This condition has two implications. On the one hand, it makes it possible to use bracketing root-finding algorithms with $\phi = [0, \theta_p]$ as initial condition. Brent's method [24] has been

2 For the sake of reproducibility, this dataset is made available online along with this article.

used to solve this equation because it is considered a fast and robust algorithm that guarantees the convergence of the solution. On the other hand, for this initial condition, the negative value of the function is associated to the lower bound of the bracket, and the positive one to the upper bound, so the resulting root represents a stationary point that turns a negative slope into a positive one. That is, this initial condition guarantees the convergence to a minimum.

4.5.2 *Numerical Results and Performance Analysis*

4.5.2.1 *Object Selection Problem*

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

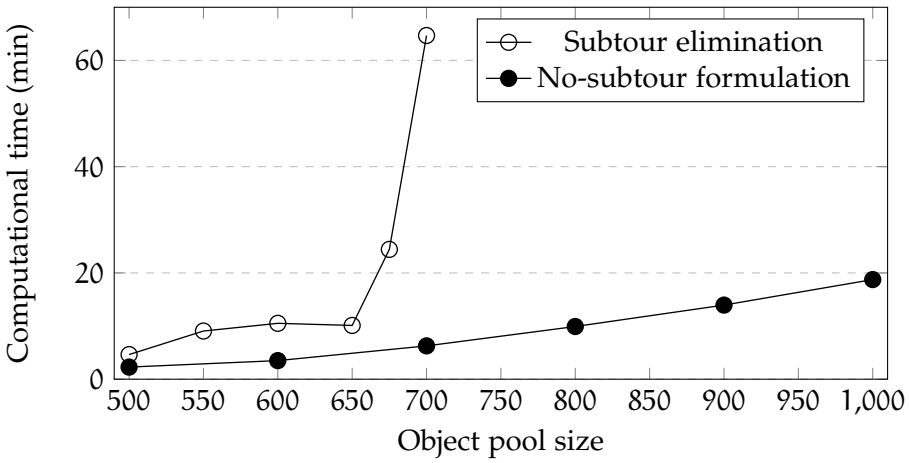


Figure 11: Comparison of the 'subtour elimination' and 'no-subtour' formulations of the object selection problem.

If a Δv budget of 6 km/s is set for this test, Figure 11 shows the computational time spent in solving the object selection problem for a growing pool of candidate objects extracted from the Test Case 1 dataset. The new, no-subtour formulation clearly outperforms the classical subtour elimination strategy and presents a very good scalability with an increasing number of candidate objects in the pool.

The disadvantage of the no-subtour formulation is that it requires a substantially larger number of variables compared to the classical

subtour elimination strategy as illustrated in Table 16. As a consequence, it exhibits more demanding memory requirements, although in practice the memory usage of the no-subtour formulation is affordable for average modern computers. This is illustrated in Table 17, which summarises the relevant metrics related to the computational cost for the scenario described in Test Case 1. At the light of this result, the no-subtour formulation is chosen for the resolution of the object selection problem in all subsequent tests.

Table 16: Comparison of object selection models: Summary of required computational resources

	Subtour elimination	**No-subtour formulation**
Constraints	$\|D\|^2 + 3\|D\| + 1$	$\|D\|^2 + (4\|K\| - 1)\|D\| - 2\|K\| + 4$
Variables	$\|D\|^2 + \|D\|$	$\|K\|\|D\|^2 + \|D\|$
Subtour elimination constraints	$2^{\|D\|} - \frac{\|D\|^2}{2} - \frac{\|D\|}{2} - 2$	0

Table 17: Comparison of object selection models: Test Case 1

	Subtour elimination	**No-subtour formulation**
Constraints	1,003,001	1,038,984
Variables	1,001,000	10,001,000
Subtour elimination constraints	$\sim 10^{301}$	0
Memory usage (MB)	628	4,960

Another interesting aspect is the quantification of the actual computational cost, which is directly related to the Δv budget. In this regard, Figure 12 displays the computational time spent for the resolution of the object selection problem with the complete pool of 1000 objects in Test Case 1, and an increasing Δv budget. It can be seen that the computational time increases with the available Δv budget, since the number of possible object combinations and sequences that need to be evaluated grows accordingly.

4.5.2.2 *Performance Assessment of the Complete Algorithm*

Once the no-subtour formulation has been determined to be a preferable option for the object selection problem (due to its superior scalability with the object pool size), the performance of the complete algorithm can now be assessed. Figure 13 shows the computational time spent in the resolution of the complete algorithm for the scenario defined in Test Case 1 for a growing Δv budget, which complements Figure 12 by displaying the total computational time, besides

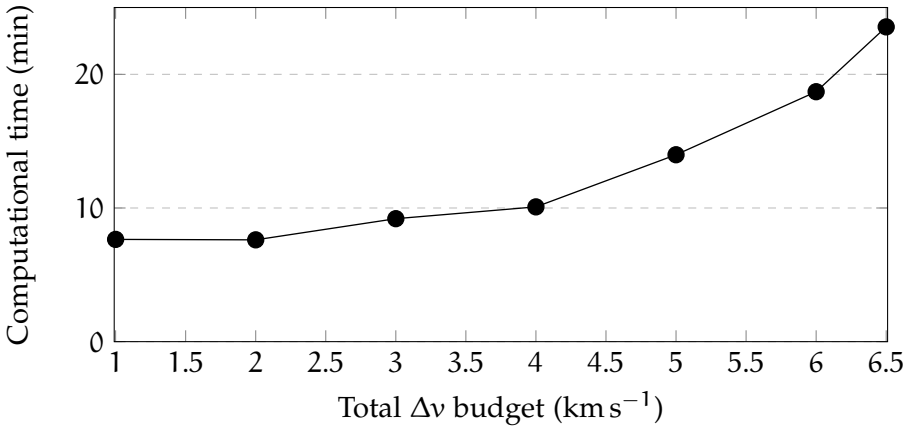


Figure 12: Computational cost of the object selection problem as a function of the Δv budget.

the time spent strictly in the object selection problem. All cases have been solved in under 10 hours, which is a satisfactory result given the complexity of the problem. However, it is worth noting that the computational time of the complete algorithm does not behave monotonically with the Δv, as opposed to the computational time for the object selection problem alone. The reason for this is that the computational time necessary to solve the complete algorithm is heavily dependent on the data and the search strategy. That is, the faster that high quality solutions are found, the faster the search space is pruned out; however, if numerous mediocre solutions are found, then the search space is pruned out slowly and a large number of feasible solutions have to be explored. This phenomenon is common for all branch-and-bound-based methods, including the object selection tests. However, because of the fact that at every time a new feasible solution is found the feasibility problem has to be solved, this effect becomes particularly meaningful.

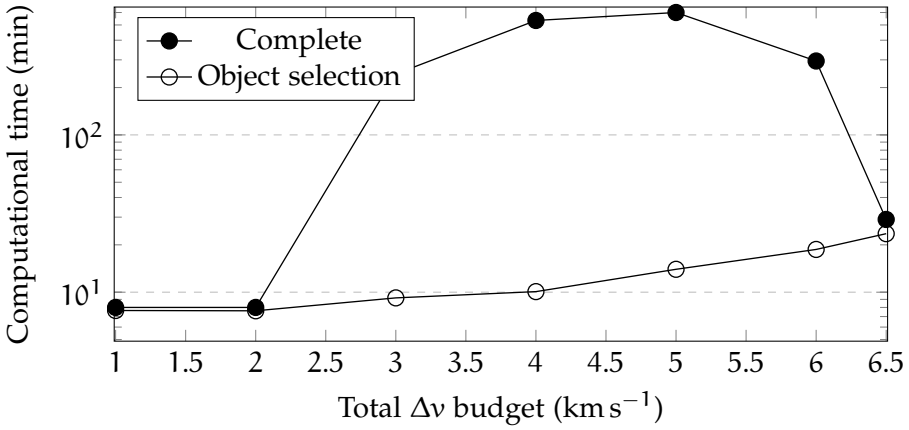


Figure 13: Computational cost for the complete algorithm and the object selection algorithm for Test Case 1.

Also, special attention has to be paid to cases with similar computational times for both tests, i.e. cases with a Δv budget of 1, 2 and 6.5 $\mathrm{km\,s^{-1}}$ respectively. In order to understand this phenomenon, Table 18 shows the obtained reward for each test. It can be seen that the optimal solutions obtained in both tests for the three considered cases are the same. As a result, they represent limit cases of the influence of data and search strategies in the computational time necessary to solve this problem. That is, the optimal solution is easily found and the remaining solutions are efficiently pruned. In fact, their solution could be obtained simply by solving the object selection problem and running the feasibility problem for its optimal solution.

Table 18: Reward comparison

Δv budget Δv_T ($\mathrm{km\,s^{-1}}$)	**1**	**2**	**3**	**4**	**5**	**6**	**6.5**
Object selection reward	99	241	414	749	870	915	925
Complete algorithm reward	99	241	382	721	866	909	925

4.5.2.3 *Performance Assessment for a Realistic Mission Scenario*

In this section, an analogous analysis to the one performed in the preceding section is carried out using the Test Case 2 dataset. It has to be noted that the injection of the satellite in the orbit of the first object to be removed is dependent on the launcher specifications and whether it is a dedicated launch, among other factors; hence, for the resolution of this case, it is considered that the Δv consumption and mission duration start counting when the satellite is in close operations with the first object to be removed. Table 19 depicts the results achieved by both, the object selection problem and the complete algorithm, for different Δv budgets.

Table 19: Summary of results for Test Case 2.

	Object selection problem			**Complete algorithm**		
Δv budget ($\mathrm{km\,s^{-1}}$)	Total criticality	Selected objects	CPU time (s)	Total criticality	Selected objects	CPU time (s)
1	9.80	2	43	9.80	2	43
2	27.46	6	42	27.46	6	200
3	44.88	9	51	44.63	8	1344
4	55.97	10	46	52.29	10	3523
5	56.98	10	69	56.98	10	4715
6	60.31	10	729	59.15	10	19504
6.5	64.08	10	125	63.13	10	11887

The object selection problem shows the total criticality that would be removed if there existed opportunities to perform perfect Hohmann transfers with optimal split plane changes between each pair of objects. However, in general it is not possible to have such transfer opportunities available within the imposed mission timeframe. As a result, the performed manoeuvres are less Δv-efficient, and thus the removed criticality obtained when solving the complete algorithm is equal to or less than the one achieved by object selection problem.

Table 20: Removal sequence for Δv budget of 3 $\mathrm{km\,s^{-1}}$

NORAD ID	Criticality Index	Δv estimation ($\mathrm{m\,s^{-1}}$)	Spent Δv ($\mathrm{m\,s^{-1}}$)
6125	5.54	-	-
15006	4.32	419.22	422.09
6993	5.48	87.34	104.52
12644	4.55	1141.48	1142.43
17536	5.89	700.33	720.95
6853	4.72	61.62	78.91
17535	13.34	282.72	283.10
6319	0.78	227.41	230.46
TOTAL	**44.63**	**2920.11**	**2982.45**

Table 21: Removal sequence for Δv budget of 4 $\mathrm{km\,s^{-1}}$

NORAD ID	Criticality Index	Δv estimation ($\mathrm{m\,s^{-1}}$)	Spent Δv ($\mathrm{m\,s^{-1}}$)
21108	2.68	-	-
20721	4.98	339.85	345.40
6125	5.54	325.70	333.45
15006	4.32	419.22	422.83
6993	5.48	87.34	117.98
12644	4.55	1141.48	1143.80
6853	4.72	726.51	747.83
17536	5.89	61.62	94.14
6319	0.78	126.01	198.83
17535	13.34	227.41	257.14
TOTAL	**52.29**	**3455.13**	**3661.40**

Tables 20 and 21 depict the removal sequences selected for the instances with a Δv budget of 3 and 4 $\mathrm{km\,s^{-1}}$, respectively. They show

the Δv consumption estimated via Hohmann transfers with optimal split plane changes (assumed by default in the object selection problem), and the actual Δv expenditure as computed by the feasibility problem. It can be seen that the 4 $km\,s^{-1}$ case (Table 21) allows to remove two additional objects compared to the 3 $km\,s^{-1}$ case (Table 20), but also provides a different removal sequence. For the sake of completeness, Table 22 collects the most relevant properties for the objects of the optimal removal sequence considered in Table 21; orbital properties are extracted from the SATCAT database, whereas satellite and rocket body masses are obtained from the NSSDCA database and Ref. [86], respectively. In this case, the proposed algorithm has been able to identify a promising cluster of high criticality objects within similar inclinations (around 74 deg) and altitudes of around 1500 km, that can be removed with the prescribed Δv budget. However, it has to be noted that the purpose of this test is not to claim that these removal sequences are the most impactful, which would be a bold statement given the assumptions made for the computation of their criticality indices and the simplified dynamics, but rather to show that the proposed methodology achieves consistent results when solving problems with a pool of realistic and heterogeneous candidate objects.

Table 22: Selected object properties

NORAD ID	Name	Perigee altitude (km)	e	i (deg)	RCS (m^2)	Mass (kg)
21108	SL-8 R/B	1459	0.0176	74.05	5.33	1435
20721	SL-14 R/B	1483	0.0025	73.60	4.79	1407
6125	SL-8 R/B	1490	0.0070	74.01	5.98	1435
15006	SL-8 R/B	1471	0.0133	74.01	5.92	1435
6993	SL-8 R/B	1475	0.0093	74.03	6.00	1435
12644	SL-8 R/B	1463	0.0130	74.03	5.66	1435
6853	SL-8 R/B	1485	0.0087	74.00	5.34	1435
17536	SL-14 R/B	1478	0.0027	73.61	5.69	1407
6319	COSMOS 539	1342	0.0023	74.01	2.29	600
17535	COSMOS 1823	1478	0.0029	73.60	12.82	700

4.6 CONCLUSIONS

This work presents a novel branch-and-bound-based algorithm that, given a large set of candidate spaceborne objects with an associated threat value, selects a subset of these objects to be removed, and defines the trajectory that allows to rendezvous with them in an opti-

mal order, so that the aggregated threat value of the removed objects is maximised, while a limit mission duration and a Δv budget are imposed as constraints.

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

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

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

This methodology has been tested using two test cases. Test Case 1 assesses the performance of the method with a 1000 object candidate pool, while Test Case 2 evaluates the behaviour of the method with a more realistic object distribution. The problem has been solved in satisfactory computational times and a heavy dependence on the search strategy has been observed. This arises the opportunity to further accelerate the problem resolution with a careful selection of the search strategy.

5

INTEGRATED MODEL FOR ACTIVE DEBRIS REMOVAL MISSIONS

5.1 INTRODUCTION

The preliminary design of multi-target active debris removal missions involves the consideration of concepts of diverse nature. On the one hand, the objects to be removed have to be determined so that the mission has as much impact as possible in the space environment. This, along with the order in which those objects are removed, constitute decisions of combinatorial nature. On the other hand, the maneuvers to perform the removal operations have to be designed so as to make the most out of the limited resources available for the development of the mission. Both the combinatorial and the dynamics-related concepts are heavily intertwined, resulting in a problem of a considerable complexity.

Focusing on the exact optimization methods to solve this kind of problems, i.e., methods that intend to mathematically demonstrate the optimality of the obtained solutions, two different kinds of approaches can be seen in the literature. Namely, a two phase approach [5, 8, 18, 30] and a sampling of the dynamics [45, 77, 85]. The two phase approach separates the dymamics from the combinatorial decisions, generating two subproblems. This way, both subproblems share information during the resolution process until the optimal solution is achieved. In turn, the other approach samples the dynamics to generate a grid of points, thus integrating the whole problem into a single combinatorial problem.

Both approaches have advantages and disadvantages, e.g., the grid approach tends to be faster and has a more global point of view of the dynamics during the resolution process, while the two phase approach considers a more accurate dynamics, tends to be much more memory-efficient and is able to exploit local properties of the function during the resolution process.

This work intends to exploit properties of both methods by means of using well known Operations Research techniques. Specifically, the Δv spent during the transfers between the objects to be removed is a function of the initial transfer time and the transfer duration. That function is discretized and approximated by a bidimensional piecewise-linear function modelled with Special Ordered Sets [82]. This results is a single MILP model that is able to obtain the global solution of the problem, according to the precission of the piecewise-

linear approximation [16], and is able to exploit the order relations between the different grid points during the branching process.

The remainder of this manuscript is organized as follows. Section 2 describes the problem at hand and presents the MILP formulation used to model it. Section 3 specifies the numerical experiments used to test the performance of the proposed methodology. Section 4 shows the results obtained after solving the aforementioned numerical experiments. Finally, Section 5 summarizes the main conclusions of this work.

5.2 PROBLEM FORMULATION

The problem at hand involves the selection of a subset of objects, from a candidate object pool, and the sequence in which they will be visited by a servicing satellite. This way, the servicing satellite will install a deorbiting kit to each of them, which, in turn, will carry out the disposal of the selected objects. Moreover, the objects have to be selected such that the aggregated criticality value assigned to each of them is maximized. In addition, the servicing satellite has to be able to visit all the selected objects within a maximum mission time and with a limited Δv budget. Hence, the problem comprises two distinct parts. Namely, the combinatorial decisions and the orbital mechanics. Both parts have been integrated into a single MILP model so that the global optimum of the problem can be obtained.

5.2.1 *Object and sequence selection*

It is clear that the object removal sequence has to form a continuous path of maneuvers that crosses all the selected objects. However, classic MILP formulations of this kind of problems cannot avoid the appearance of disjoint loops called *subtours*. The conventional way to address this problem is to dynamically generate *subtour* elimination constraints each time a new *subtour* appears [39]. In general, this is a very efficient technique, as the number of relevant *subtours* tends not to be very large. Nevertheless, this is not true for problems comprising numerous objects distributed among diverse orbital planes. This happens because the appearance of *subtours* allows to artificially substitute expensive change of plane maneuvers with cheaper ones. Thus generating a great amount of invalid solutions. In order to avoid this phenomenon, the formulation in Ref. [8] prevents the appearance of *subtours*. The objective function of the problem and the constraints that guarantee the formation of a continuous path are subsequently introduced.

The objective function maximizes the aggregated criticality value of the removed objects:

$$\max \left\{ \sum_{i \in D} R_i \, Y_i \right\} \tag{116}$$

where the set D (indexed by i and j) represents the candidate object pool, the binary variable Y_i is 1 if the object i is selected and is 0 otherwise and the parameter R_i is the criticality index of object i.

The servicing satellite cannot perform a round trip journey between two objects:

$$\sum_{k \in K} X_{ijk} + \sum_{k \in K} X_{jik} \leqslant 1 \quad \forall i \in D, \forall j \in D : (i \neq j) \tag{117}$$

where the set K (indexed by k) represents the maneuver sequence and the binary variable X_{ijk} is 1 if the k^{th} transfer happens from object i to object j and is 0 otherwise.

It is not possible to perform a transfer between an object and itself:

$$X_{iik} = 0 \quad \forall i \in D, \quad \forall k \in K \tag{118}$$

Each object is associated to, at most, one incoming and one outgoing transfer:

$$\sum_{\substack{j \in D \\ i \neq j}} \sum_{k \in K} X_{ijk} + \sum_{\substack{j \in D \\ i \neq j}} \sum_{k \in K} X_{jik} \geqslant Y_i \quad \forall i \in D \tag{119}$$

$$\sum_{\substack{j \in D \\ i \neq j}} \sum_{k \in K} X_{ijk} + \sum_{\substack{j \in D \\ i \neq j}} \sum_{k \in K} X_{jik} \leqslant 2Y_i \quad \forall i \in D \tag{120}$$

The number of removed objects is the number of performed transfers plus 1:

$$\sum_{i \in D} \sum_{\substack{j \in D \\ i \neq j}} \sum_{k \in K} X_{ijk} = \sum_{i \in D} Y_i - 1 \tag{121}$$

Each position has associated, at most, one transfer:

$$\sum_{i \in D} \sum_{\substack{j \in D \\ i \neq j}} X_{ijk} \leqslant 1 \qquad \forall k \in K \tag{122}$$

The final object of a transfer has to be the first of the subsequent one:

$$\sum_{\substack{i \in D \\ i \neq j}} X_{ijk} \geqslant \sum_{\substack{i \in D \\ i \neq j}} X_{ji(k+1)} \qquad \forall j \in D, \quad \forall k \in K : (k < |K|) \tag{123}$$

5.2.2 *Piecewise-linear dynamics*

The Δv consumed during a transfer is a function of its starting time as well as its duration. In general, this function is nonlinear and non-convex, which integrated with the combinatorial model would result in a Mixed Integer Nonlinear Programming problem. Such problems are difficult to solve, especially when trying to compute the global optimum. However, if a piecewise-linear approximation of the Δv is used instead, the resulting model will constitute a MILP problem. A classic technique to develop these approximations relies on the concept of Special Ordered Sets (SOS) [16]. As its name implies, a Special Ordered Set is an ordered set of variables whose order properties can be exploited with specific branching techniques. There are two kinds of SOS, namely, SOS1 and SOS2. They differ in the fact that at most one variable pertaining to a SOS1 set can have a nonzero value, while SOS2 sets accept up to two nonzero variables as long as they are consecutive.

The methodology presented in [82] makes use of both kinds of SOS to develop a bidimensional piecewise-linear approximation. Its has been applied to the problem at hand as follows:

- The function domain is discretized into a grid of nodes. In this particular case, the nodes $b \in B$ represent the transfer starting times, while $c \in C$ stand for values of the transfer duration.
- Two sets of SOS1 variables (λ_b, μ_c) are used to define the sides of rectangular boxes formed by adjacent nodes.
- A set of SOS2 variables (η_s) is used to divide each rectangle into two triangles, where $s \in S$ is the set of parallel diagonals of the whole domain.
- Weight variables (w_{bc}) are computed according to the values of the SOS variables. The product of those weights and the Δv values on the grid points provides the desired Δv approximation.

The computation of the weight variables can be modelled with a set of constraints. The following constraints impose that, for a rectangular box defined by the values of λ_b and μ_c, only the weights corresponding to the vertices of that box can be nonzero.

$$\sum_{c \in C} w_{1c} \leqslant \lambda_2 \tag{124}$$

$$\sum_{c \in C} w_{bc} \leqslant \lambda_b + \lambda_{(b+1)} \quad \forall b \in [2, |B| - 1] \tag{125}$$

$$\sum_{c\in C} w_{|B|c} \leqslant \lambda_{ij|B|} \tag{126}$$

$$\sum_{b\in B} w_{b1} \leqslant \mu_2 \tag{127}$$

$$\sum_{b\in B} w_{bc} \leqslant \mu_c + \mu_{(c+1)} \quad \forall c \in [2, |C| - 1] \tag{128}$$

$$\sum_{b\in B} w_{b|C|} \leqslant \mu_{|C|} \tag{129}$$

Furthermore, only the weights belonging to two consecutive diagonals can be nonzero:

$$\eta_s = \sum_{b\in B'(s)} w_{b(b-|B|+s)} \quad \forall s \in S \tag{130}$$

where the set $B'(s)$ is defined as:

$$B'(s) = b \in B : \max(1, |B| + 1 - s) \leqslant b \leqslant \min(|B| + |C| - s, |B|) \tag{131}$$

This way, Eqs. (124-130) ensure that only the weights corresponding to the triangle selected by the SOS sets can be nonzero. Finally, if the sum of each of the SOS sets is 1, so is the sum of the weight variables:

$$\sum_{\substack{b\in B \\ b>1}} \lambda_b = 1 \tag{132}$$

$$\sum_{\substack{c\in C \\ c>1}} \mu_c = 1 \tag{133}$$

$$\sum_{s\in S} \eta_s = 1 \tag{134}$$

5.2.3 *Integrated formulation*

The integration of the piecewise-linear dynamics with the object and sequence selection requires additional considerations. Specifically, Eqs. (116-123) can be readily introduced in the integrated model. However, the piecewise-linear dynamics model has to be applied for each ij object

combination. This implies that the variables λ_b, μ_c, η_s and w_{bc} have to be substituted with their generalized counterparts, i.e., $\lambda_{ijb}, \mu_{ijc}, \eta_{ijs}$ and w_{ijbc}, in Eqs. (124-130). Moreover, it is necessary to compute the total Δv spent during the mission, as well as the starting time and duration of each of the transfers as a function of the w_{ijbc} weights. Those quantities can be readily determined by the following equations:

$$\sum_{i\in D}\sum_{\substack{j\in D\\ i\neq j}}\sum_{b\in B}\sum_{c\in C}\Delta v^{fun}_{ijbc}w_{ijbc} \leqslant \Delta v^{tot} \tag{135}$$

$$\sum_{k\in K} t_{ijk} = \sum_{b\in B}\sum_{c\in C} t^{fun}_{ijb}w_{ijbc} \quad \forall i \in D, \forall j \in D : (i \neq j) \tag{136}$$

$$\sum_{k\in K} \Delta t_{ijk} = \sum_{b\in B}\sum_{c\in C} \Delta t^{fun}_{ijc}w_{ijbc} \quad \forall i \in D, \forall j \in D : (i \neq j) \tag{137}$$

where Δv^{tot} is the Δv budget and the parameter Δv^{fun}_{ijbc} represents the Δv values in each of the grid nodes associated to each of the possible transfers. Likewise, t^{fun}_{ijb} and Δt^{fun}_{ijc} are homologous parameters representing the starting times of the transfers and their durations, respectively.

Unfortunately, the simple introduction of Eqs. (135-137) does not ensure the correct computation of the aforementioned quantities. This is due to the fact that even when a transfer between i and j is not performed in the selected sequence, nonzero weights can be assigned to it. Hence generating spurious contributions in the summations of Eqs. (135-137). Nevertheless, this problem can be solved by means of substituting Eqs. (132-134) with the following ones:

$$\sum_{\substack{b\in B\\ b>1}} \lambda_{ijb} = \sum_{k\in K} X_{ijk} \quad \forall i \in D, \forall j \in D : (i \neq j) \tag{138}$$

$$\sum_{\substack{c\in C\\ c>1}} \mu_{ijc} = \sum_{k\in K} X_{ijk} \quad \forall i \in D, \forall j \in D : (i \neq j) \tag{139}$$

$$\sum_{s\in S} \eta_{ijs} = \sum_{k\in K} X_{ijk} \quad \forall i \in D, \forall j \in D : (i \neq j) \tag{140}$$

This way, the SOS variables vanish for the transfers not included in the selected sequence and Eqs. (124-130) force their associated weights to be null. However, it has to be noted that Eqs. (136,137) deal with $\sum_{k\in K} t_{ijk}$ and $\sum_{k\in K} \Delta t_{ijk}$ instead of the straightforward

variables. Therefore, those equations are not able to unambiguously determinate t_{ijk} and Δt_{ijk}, as they do not discriminate between the different values of k. The following equations make use of the information contained in X_{ijk} to eliminate this ambiguity.

$$t_{ijk} \geqslant t^{fun}_{ij1} X_{ijk} \quad \forall i \in D, \forall j \in D : (i \neq j), \forall k \in K \tag{141}$$

$$t_{ijk} \leqslant t^{fun}_{ij|B|} X_{ijk} \quad \forall i \in D, \forall j \in D : (i \neq j), \forall k \in K \tag{142}$$

$$\Delta t_{ijk} \geqslant \Delta t^{fun}_{ij1} X_{ijk} \quad \forall i \in D, \forall j \in D : (i \neq j), \forall k \in K \tag{143}$$

$$\Delta t_{ijk} \leqslant \Delta t^{fun}_{ij|C|} X_{ijk} \quad \forall i \in D, \forall j \in D : (i \neq j), \forall k \in K \tag{144}$$

Therefore, it is guaranteed that, if the k^{th} transfer of the sequence happens between the objects i and j, the variables t_{ijk} and Δt_{ijk} represent the starting time and duration or that transfer, respectively, and otherwise both variables vanish.

Furthermore, the close operations to eliminate all the selected objects have to be completed within the maximum mission time:

$$\sum_{i \in D} \sum_{\substack{j \in D \\ i \neq j}} \left(t_{ijk} + \Delta t_{ijk} + \Delta t^{ren} X_{ijk} \right) \leqslant t^{max} \quad \forall k \in K \tag{145}$$

where Δt^{ren} is the time to complete the close operations and t^{max} is the maximum mission time.

Finally, the starting times of each of the maneuvers have to follow a logical sequence, i.e., a transfer that happens right after another has to start at time greater than the time in which the close operations with the previous object have finished:

$$\sum_{i \in D} \sum_{\substack{j \in D \\ i \neq j}} t_{ij(k+1)} \geqslant \sum_{i \in D} \sum_{\substack{j \in D \\ i \neq j}} \left(t_{ijk} + \Delta t_{ijk} + \Delta t^{ren} X_{ijk} \right) - t^{max} \left(1 - \sum_{i \in D} \sum_{\substack{j \in D \\ i \neq j}} X_{ij(k+1)} \right) \quad \forall k \in K : (k < |K|) \tag{146}$$

where the last term of the right hand side makes sure that this constraint remains feasible when no objects are eliminated in the $k+1$ position.

5.3 NUMERICAL EXPERIMENTS DEFINITION

Meaningful and realistic numerical experiments have been configured so as to test the performance of the proposed methodology. First, the considered values of the operational constraints are the following:

- Δv budget (Δv^{tot}): 2000 ms^{-1}.
- Maximum mission time (t^{max}): 1 year.
- Time to complete the close operations (t^{ren}): 1 day.

The candidate object pool has to comprise a significant number of high-priority pieces of debris. A ranking of the 70 most relevant pieces of debris in LEO, according to a particular criticality index, can be found in [23]. Two prominent object clusters can be easily identified. Specifically, a group of 30 objects with inclinations close to 98 degrees and other of 36 objects with inclinations close to 71 degrees. A test case has been configured for each of those clusters.

Moreover, the size of the candidate object pool ($|I|$) has a great impact in the model size and, consequently, it is expected to have a great influence on the computational time necessary to solve the problem. Hence, it is interesting to test the influence of the least important objects of each of the pools both on the computational time and the quality of the solution. To that end, such clusters can be further partitioned according to the object criticality. This way, a subset of 15 objects from the 98-degrees-of-inclination cluster with criticality index greater than 10, as well as two subsets of sizes 16 and 26, and criticality indices greater than 40 and 10, respectively, from the 71-degrees-of-inclination cluster are also evaluated. On a similar note, the maximum number of transfers that can be performed ($|K|$) also has a remarkable impact in the model size. Therefore, different values of this parameter are tested for each problem instance.

Finally, the Δv consumption as a function of the starting time and transfer duration has to be defined. Regarding the orbits of the objects as circular, Ref. [94] provides an analytical approximation for the Δv consumed during a low-thrust transfer under J_2-perturbed dynamics. As it is expected that the objects within one of the aforementioned clusters will have similar nodal precessions, the Δv consumption will be less sensitive to the starting time than the transfer duration. Thus, only 2 starting time nodes ($|B|$) will be considered, along with 9 transfer duration nodes ($|C|$).

5.4 RESULTS

The proposed formulation has been implemented in GAMS 25.1.1 and the aforementioned test cases have been solved with CPLEX 12.8.0 in a PC featuring an Intel Core i7-1165G7 and 16 GB of RAM.

5.4.1 *Test case 1 (98-degrees-of-inclination cluster)*

Table 23 shows the criticality and computational times obtained when solving Test Case 1 for different values of |I| and |K|. It can be seen that those two values have a remarkable impact in the computational time. Interestingly, using the previously mentioned subsets of the object pool results in the solutions obtained with the complete object pool, but with significantly smaller computational times. In turn, the increment of |K| results in a greater computational time and a better solution quality up to a point in which the Δv and mission time constraints make it impossible to remove additional objects. This way, the case with $|K| = 8$ and $|I| = 15$ provides the same solution as the one with $|K| = 7$ and $|I| = 15$, but at a greater computational cost.

Table 24 shows the best removal sequence for this test case. It removes eight objects, including rocket bodies as well as payloads, with the Envisat being the most notable of them.

Table 23: Summary of results for Test Case 1.

|I|	|K|	Objects removed	Criticality	CPU time (s)
15	5	6	163.91	26
15	6	7	174.54	291
15	7	8	186.63	347
15	8	8	186.63	2791
30	5	6	163.91	326
30	6	7	174.54	1073
30	7	8	186.63	10871

Table 24: Test Case 1 optimal removal sequence.

NORAD ID	Name	Semimajor axis (km)	Inclination (deg)	Criticality
25400	SL-16 R/B	7185	98.64	41.38
37932	CZ-2D R/B	7196	98.69	13.77
27597	ADEOS 2	7178	98.57	12.14
33272	COSMOS 2441	7098	98.11	23.94
27386	ENVISAT	7143	98.16	50.43
28480	CZ-2C R/B	7184	98.05	12.69
44548	CZ-2D R/B	7139	98.14	15.28
41858	CZ-2D R/B	7150	98.51	17.00

5.4.2 *Test Case 2 (71-degrees-of-inclination cluster)*

Table 25 shows the results obtained for Test Case 2. They show a behavior similar to the one observed for Test Case 1. It leads to think that it is reasonable to exclude from the candidate pool objects with modest criticality indices. Special attention has to be paid to the case with $|K| = 7$ and $|I| = 16$. After 4 hours of computational time, no solutions better than the one obtained for the case with with $|K| = 6$ and $|I| = 16$ where found, but the optimality of this solution was not demonstrated, just that the criticality of the optimal solution has to be contained within the interval $[313.98, 342.57]$. However, there is an alternative approach to obtain this solution. Specifically, instead of looking for the optimal solution that removes up to 8 objects, one can try to demonstrate the feasibility of removing 8 objects. As the worst aggregated criticality value that can be achieved by an 8-object sequence is 335.24, this alternative approach simply requires pruning any branch of the search tree with a potential criticality value lower than that. This way, it takes less than 3 minutes to demonstrate the infeasibility of performing an 8-object sequence for this test case.

Table 26 shows the best removal sequence for Test Case 2. It removes seven objects, as opposed to the eight removed in Test Case 1. However, all of the objects removed in this case show a remarkably large criticality. Thus resulting in an aggregated criticality value much larger than the one obtained in Test Case 1. It has to be noted that all of the objects removed in Test Case 2 are rocket bodies.

Table 25: Summary of results for Test Case 2.

$\|I\|$	$\|K\|$	Objects removed	Criticality	CPU time (s)
16	5	6	273.23	205
16	6	7	313.98	347
16	7	7	[313.98,342.57]	14400
16	7	8	Infeasible	164
26	5	6	273.23	396
26	6	7	313.98	1159
36	5	6	273.23	542
36	6	7	313.98	1954

5.5 CONCLUSIONS

This manuscript proposes a novel MILP-based approach for the global optimization of multi-target active debris removal missions. Specifically, the proposed formulation involves combinatorial decisions (tar-

Table 26: Test Case 2 optimal removal sequence.

NORAD ID	Name	Semimajor axis (km)	Inclination (deg)	Criticality
25407	SL-16 R/B	7218	71.01	42.07
23705	SL-16 R/B	7220	71.03	42.51
31793	SL-16 R/B	7222	70.98	51.69
19650	SL-16 R/B	7218	71.00	42.07
16182	SL-16 R/B	7216	71.00	41.97
26070	SL-16 R/B	7219	71.00	49.40
23088	SL-16 R/B	7221	71.00	44.27

get and sequence selection) as well as dynamics-related variables (the Δv spent during each of the transfers as a function of their starting time and duration). The complexity of the combinatorial decisions has been addresed by means of the use of a no-subtour MILP formulation. In turn, the nonlinearity, nonconvexity and nonstationarity of the orbital mechanics has been tackled with a piecewise-linear approximation modelled with Special Ordered Sets. Both techniques have been integrated into a single MILP model whose resolution provides a global optimum of arbitrary precision, i.e., according to the number and position of the nodes used to constitute the piecewise-linear approximation.

The performance of the methodology has been assessed using two candidate object pools. In particular, two clusters of high priority targets in LEO with inclinations around 71 and 98 degrees, respectively. Optimal removal sequences involving up to seven objects for the 71 degrees of inclination case and up to eight objects for the 98 degrees case have been obtained in very affordable computational times.

6

ANALYSIS OF A FUEL STATION STRATEGY FOR DEBRIS REMOVAL

6.1 INTRODUCTION

It is expected that active space debris removal missions will remove several objects with each of the employed servicing satellites. Moreover, the larger the mass of a servicing satellite, the larger the fuel consumed during a maneuver with a given Δv. This way, the removal of a considerable number of objects with a single satellite can result in an inefficient use of resources. A mission architecture that can potentially ameliorate this effect involves the use of a fuel station that the servicing satellite can visit to refuel itself. Hence, making it possible to perform the required maneuvers with a lighter servicing satellite.

This concept has been thoroughly explored for satellite refuelling missions in geosynchronous orbit. Zhou et al.[112] propose a methodology to solve refuelling missions that consider one servicing satellite and a single fuel station with a predefined initial position. In a subsequent contribution, Zhou et al.[113] extend the scope of the problem by means of using multiple servicing satellites and fuel stations. Zhang et al.[110] consider multiple servicing satellites and a set of potential locations in which fuel stations can be allocated, thus introducing an additional decision variable to the problem. Zhu et al.[114] use clustering techniques to determine the location of the fuel stations as a function of the positions of the targets.

Furthermore, in the scope of active debris removal missions, this concept has been proposed as a possible architecture for the removal of failed satellites of large constellations in Low Earth Orbit (LEO). Colombo et al.[35] propose a mission comprising a servicing satellite and a fuel station. In particular, the servicing satellite transports the defunct satellites to disposal orbits compliant with debris mitigation guidelines and visits the fuel station to refuel itself after the removal of each object.

The mission considered in this work involves the selection of a set of pieces of debris in LEO to be removed. This way, the selected targets will be visited by a servicing satellite, which will attach deorbiting kits to each of them. Moreover, the servicing satellite can visit a fuel station to refuel itself as well as to load the deorbiting kits to be attached to the subsequent batch of targets. Hence, the initial position of the station has to be selected so as to obtain the most advantageous mission.

This work proposes a two-stage framework to solve the aforementioned problem. Specifically, the upper stage determines the location of the fuel station, while the lower stage selects the objects to be removed and optimises the manoeuvres the servicing satellite has to perform. This way, both stages are iteratively solved until the desired solution is found. Moreover, a heuristic strategy to obtain an high-quality initial guess of the location of the fuel station has been devised, hence potentially accelerating the resolution of the problem.

6.2 PROBLEM DESCRIPTION

Given a pool of candidate objects to be removed, the problem at hand entails the selection of a subset of those objects such that the aggregated criticality value assigned to them is maximized, while complying with a maximum mission time and Δv budget constraints.

The removal is performed by a servicing satellite, which will rendezvous with each of the selected objects so as to attach deorbiting kits to them. It is considered that this spacecraft is equipped with a low-thrust propulsion system that is able to generate a certain aggregated Δv value. Moreover, the servicing satellite can rendezvous with a fuel station to reset its available Δv, as well as to reload the necessary deorbiting kits. In particular, the servicing satellite starts at a predefined initial position, will perform a fixed number of visits to the fuel station and will deorbit itself along with the final target. In turn, the fuel station is not able to perform any kind of propulsive maneuvers. As a result, the removal sequence of the selected objects, the maneuvers performed by the servicing satellite and the initial location of the fuel station are also variables of the problem.

Finally, it has to be noted that the considered dynamics assumes that the orbits of all the involved objects and spacecraft are circular and are subjected to the averaged effect of the J_2 perturbation in the Right Ascension of the Ascending Node (RAAN) as follows:

$$\dot{\Omega} = -\frac{3}{2} R_{\oplus}^2 J_2 \sqrt{\frac{\mu}{a^7}} \cos i \tag{147}$$

where $\dot{\Omega}$ is the rate of change of the RAAN, $R_{\oplus}$ is the equatorial radius of the Earth, J_2 is the coefficient of the spherical harmonic of degree 2 and order 0 of the Earth's gravity field, μ is the gravitational parameter of the Earth and a and i are the semimajor axis and the inclination of the involved object, respectively.

6.3 METHODOLOGY

The described problem involves a nonlinear dynamics and requires the determination of continuous as well as discrete variables. Hence,

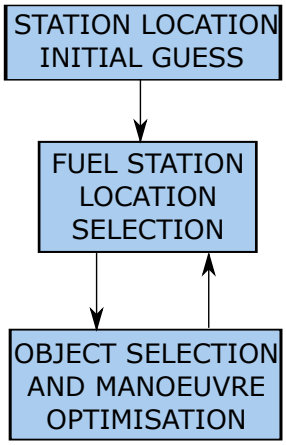


Figure 14: Problem resolution process

it can be modelled as a Mixed Integer Nonlinear Programming problem. As this kind of problem structure is generally difficult to solve, the proposed methodology divides it into two different stages involving simpler subproblems, as depicted in Figure 14.

On the one hand, the upper stage explores promising values of the station location and sends them to the lower stage. On the other hand, the lower stage determines the objects to be removed, the removal sequence and the maneuvers of the servicing satellite so that the aggregated criticality of the selected objects is maximized. In addition, a preliminary process to determine a high-quality location of the fuel station has been devised. In particular, this preliminary process intends to reduce the number of fuel station locations evaluated by the upper stage and, in some cases, it is able to readily determine a station location that results in the global optimal value of the aggregated criticality removed by the servicing satellite.

6.3.1 *Lower stage*

Given an initial position for the fuel station, the lower stage selects the targets to be removed, their associated removal order and the trajectory that the servicing satellite has to describe to successfully maximize the aggregated criticality of the selected objets. This is still a Mixed Integer Nonlinear Programming problem. Nevertheless, Special Ordered Sets can be used to approximate the nonlinearities as piecewise linear functions, thus resulting in a Mixed Integer Linear Programming (MILP) problem[82]. This way, Branch-and-Bound methods are able to obtain the global optimum of this problem, subject to the precision of the approximation of the nonlinear functions[16].

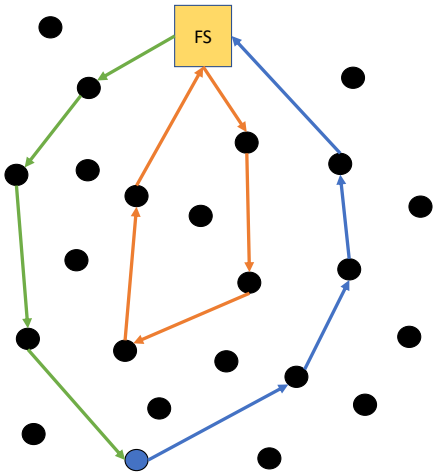


Figure 15: Removal sequence scheme

Specifically, the trajectory described by the servicing satellite follows the structure depicted in Figure 15, where the black dots represent candidate objects to be removed, the blue dot is a dummy object that represents both the initial position and disposal orbit of the servicing satellite and the yellow square stands for the fuel station. This way, such trajectory can be modelled with two rectilinear graphs between the fuel station and the dummy object, split apart by a number of cycles that contain the fuel station. Each of those graphs can be efficiently defined with a MILP formulation specifically taylored for multi-target active debris removal missions[8]. Furthermore, the MILP model of the problem at hand can be completed with a piecewise-linear approximation of the dynamics compatible with the structure of the problem [9]. This approximation of the dynamics is used to impose that each of the aforementioned graphs has to fulfil a Δv budget constraint and that the whole mission has to be completed within a maximum mission time.

6.3.2 *Upper stage*

The purpose of the upper stage is to sample the configuration space of initial locations of the fuel station. This way, the sampled positions are subsequently sent to the lower stage, thus obtaining an assessment of their quality. Then, such information is used to sample additional points until a satisfactory solution is found.

In particular, this stage can be modelled as the unconstrained optimization of a black-box function (i.e., the lower stage). As the lower stage represents a discontinuous non-smooth function, derivative-free

methods have to be used to explore the search space. Usual metaheuristic methods can readily provide an effective sampling of the search space and, consequently, a good-quality solution. However, the resolution of the lower stage can be computationally expensive, therefore, a local search method with a high-quality initial guess is proposed instead, specifically, Generalized Pattern Search[72] has been considered.

6.3.3 *Heuristic initial guess*

The determination of an initial guess for the position of the station is of great importance for the proposed methodology. In particular because a bad selection of such guess, not only might require a large number of evaluations of the lower stage until a local optimum is obtained, but also because the obtained solution might provide a disadvantageous value of aggregated criticality.

A strategy to develop such heuristic guesses is to make assumptions about properties of the optimal or near-optimal solutions, such that the fulfilment of such properties significantly simplifies the problem. In this case, the assumption is that the Δv constraint imposed to each of the graphs results in an inefficient Δv allocation, hence resulting in unspent Δv. Then, it is also assumed that such Δv surplus is enough to compensate the additional Δv consumption produced by the transfers to the station.

Therefore, if this assumption is correct, the solution obtained when solving the lower stage with a null Δv consumption in the incoming transfer to the station and a Δv consumption in the outgoing transfer equal to the one necessary to directly transfer between the two objects adjacent to the station would be the global optimum of the problem. In turn, if the assumption is not correct, such solution is an optimistic bound of the global optimum. Thus, the proposed initial guess process mirrors the structure of the aforementioned two-stage process in an attempt to determine the global optimal or near optimal solutions, but takes advantage of the heuristic assumptions to simplify the resolution of both stages. Specifically, the two iterative stages of the heuristic methodology involve the determination of the initial guess of the station location and the resolution of a modified lower stage, respectively.

Regarding the initial guess of the station location, it involves the determination of its semimajor axis, inclination and initial RAAN. First, the RAAN and RAAN drift of the fuel station that minimize the square RAAN difference with its preceding objects are determined. This can be modelled with the following objective function:

$$J = \min \sum_{\ell \in L} \left(\Omega_\ell - \Omega_s - t_\ell \dot{\Omega}_s \right)^2 \tag{148}$$

where J is the value of the objective function, the set L (indexed by ℓ) represents the different refills, t_ℓ is the starting time of the ℓ-th transfer to the fuel station, Ω_ℓ is the RAAN of the object visited before the ℓ-th refill at t_ℓ, Ω_s is the initial RAAN of the station and $\dot{\Omega}_s$ is the nodal drift of the station.

The optimality conditions of the problem can be obtained by nullifying the gradient of the objective function:

$$\frac{\partial J}{\partial \Omega_s} = -2 \sum_{\ell \in L} \left(\Omega_\ell - \Omega_s - t_\ell \dot{\Omega}_s \right) = 0 \tag{149}$$

$$\frac{\partial J}{\partial \dot{\Omega}_s} = -2 \sum_{\ell \in L} \left(t_\ell \Omega_\ell - t_\ell \Omega_s - t_\ell^2 \dot{\Omega}_s \right) = 0 \tag{150}$$

Thus, the optimal solution can be analitically obtained by isolating Ω_s and $\dot{\Omega}_s$ from Eqs. (149) and (150). However, the cases with one and two refills show interesting properties.

Regarding the case with just one refill, it can be seen that Eqs. (149) and (150) are proportional. Hence, the system of equations is indeterminate and has an infinite number of solutions. A notable solution requires that the station and the object visited before the refill have the same position throughout the whole mission (i.e., $\Omega_s = \Omega_{\ell_1}$ and $\dot{\Omega}_s = \dot{\Omega}_\ell$). Therefore, the transfer to the station is free and that solution constitutes the global optimum of the problem.

Regarding the case with two refills, the optimal solution is able to nullify the objective function in every case and is defined as follows:

$$\Omega_s = \Omega_{\ell_1} - \frac{\Omega_{\ell_2} - \Omega_{\ell_1}}{t_{\ell_2} - t_{\ell_1}} t_{\ell_1} \tag{151}$$

$$\dot{\Omega}_s = \frac{\Omega_{\ell_2} - \Omega_{\ell_1}}{t_{\ell_2} - t_{\ell_1}} \tag{152}$$

In turn, the optimal solution for an arbitrary number of refills is the following one:

$$\Omega_s = \frac{\sum_{\ell \in L} \Omega_\ell \sum_{\ell \in L} t_\ell^2 - \sum_{\ell \in L} (\Omega_\ell t_\ell) \sum_{\ell \in L} t_\ell}{|L| \sum_{\ell \in L} t_\ell^2 - \left(\sum_{\ell \in L} t_\ell \right)^2} \tag{153}$$

$$\dot{\Omega}_s = \frac{|L| \sum_{\ell \in L} (\Omega_\ell t_\ell) - \sum_{\ell \in L} \Omega_\ell \sum_{\ell \in L} t_\ell}{|L| \sum_{\ell \in L} t_\ell^2 - \left(\sum_{\ell \in L} t_\ell \right)^2} \tag{154}$$

Then, the nodal drift of the station has to be translated into a semi-major axis and an inclination. The values of those variables are selected so that they minimize the aggregated Δv consumption to transfer between the preceding objects and the station:

$$\min \sum_{\ell \in L} \Delta v_\ell = \sum_{\ell \in L} \left(\sqrt{\frac{\mu}{a_\ell} + \frac{\mu}{a_s} - 2\sqrt{\frac{\mu}{a_\ell}}\sqrt{\frac{\mu}{a_s}} \cos\left(\frac{\pi}{2}(i_s - i_\ell)\right)} \right) \quad (155)$$

where the Δv_ℓ consumption has been modelled with the Edelbaum analytical solution[41], a_ℓ and i_ℓ are the semimajor axes and inclinations of the preceding objects, respectively, and a_s and i_s are the semimajor axis and inclination of the station, respectively.

In addition, the previously computed nodal drift of the station has to comply with the averaged effect of the J_2 perturbation:

$$\dot{\Omega}_s = -\frac{3}{2} R_\oplus^2 J_2 \sqrt{\frac{\mu}{a_s^7}} \cos i_s \quad (156)$$

The solution of this problem can be readily obtained by means of isolating a_s in Eq. (156), substituting it into Eq. (155) and nullifying the derivative of Eq. (155) with respect to i_s, thus involving the resolution of just a univariate equation.

Regarding the resolution of a modified lower stage, it simply involves solving such problem disregarding the existence of the fuel station and imposing minimum values of duration and Δv for the transfers between the objects that would be adjacent to the station. Those values have been computed in a previous iteration of the initial guess obtention and stand for the sum of the duration and Δv spent during the corresponding incoming and outgoing transfers between the involved objects and the station.

This way, the algorithm of the initial guess obtention operates as follows:

1. Solve the modified lower stage for null minimum time and Δv.
2. Store the optimistic bound of the problem.
3. Compute the station location associated to that solution.
4. Compute the transfer duration and Δv associated to that station location.
5. Solve the modified lower stage with minimum values equal to the transfer duration and Δv and compute the optimality gap.
6. If the aggregated criticality value is equal to the one of the previous modified lower stage, exit.
7. Else, repeat Steps 3 to 7 until the optimality gap is greater than a threshold.

The aforementioned optimality gap quantifies the relative difference of the obtained criticality value with respect to the optimistic

bound. Therefore, if the optimality gap is less than the predefined threshold, it is considered that the quality of the solution is acceptable and it is not necessary to execute the loop corresponding to the original two-stage process. Otherwise, such loop will be executed with the most recent value of the station location as the initial guess.

6.4 RESULTS

Two practical cases involving debris clusters of high-impact LEO objects with inclinations around 71 and 98 degrees, respectively, are subsequently analyzed. In particular, criticality values previously computed in the literature [23] have been assigned to the concerning objects.

For each of those cases, instances with one and two refills have been solved for a servicing satellite with 1 km/s of Δv capacity, as well as two additional instances without refills, but with a Δv capacity of 2 and 3 km/s, respectively.

A maximum mission time of one year has been imposed for all the cases and low thrust maneuvers have been considered, Specifically, the maneuvers are modelled with the Split Edelbaum Strategy[32] or the Relative Inclination Change strategy [12], whichever achieves a lower Δv for each transfer duration and starting time.

6.4.1 *Test Case 1 (71 degrees of inclination)*

The Test Case 1 involves 36 objects with an inclination around 71 degrees. Table 27 summarizes the results of the four instances solved for this test case. Regarding the instances with refills, the heuristic initial guess process has been able to determine the global optimality of both instances. Regarding the instances without refills, it can be seen that the instance with a capacity of 2km/s achieves the same result as the instance with one refill. In turn the instance with a capacity of 3km/s is able to remove nine objects, as opposed to the eight that the instance with two refills is able to solve. This possibly indicates a case of diminishing returns when increasing the number of refills, that is, the greater the number of refills, the less Δv-efficient the solution is.

Table 27: Results comparison for Test case 1

Refills	Vehicle capacity (m/s)	Iterations	Removed Objects	Criticality	Optimality gap (%)
2	1000	1	7	313.98	0
3	1000	2	8	357.27	0
0	2000	1	7	313.98	0
0	3000	1	9	396.83	0

Tables 28, 29 and 30 show the removal sequences of the four instances. It can be seen that the selected objects are practically the same, albeit with slightly different removal sequences. In particular, the objects, with Norad IDs 25407,23705, 31793, 19650, 16182, 26070 and 23088 are common to the four instances.

Table 28: Removal sequence for Test Case 1 with one refill

NORAD ID	Name	Semimajor axis (km)	Inclination (deg)	Criticality
25407	SL-16 R/B	7218	71.01	42.07
23705	SL-16 R/B	7220	71.03	42.51
31793	SL-16 R/B	7222	70.98	51.69
19650	SL-16 R/B	7218	71.00	42.07
16182	SL-16 R/B	7216	71.00	41.97
26070	SL-16 R/B	7219	71.00	49.40
23088	SL-16 R/B	7221	71.00	44.27

Table 29: Removal sequence for Test Case 1 with two refills

NORAD ID	Name	Semimajor axis (km)	Inclination (deg)	Criticality
25407	SL-16 R/B	7218	71.01	42.07
23705	SL-16 R/B	7220	71.03	42.51
31793	SL-16 R/B	7222	70.98	51.69
19650	SL-16 R/B	7218	71.00	42.07
16182	SL-16 R/B	7216	71.00	41.97
26070	SL-16 R/B	7219	71.00	49.4
23088	SL-16 R/B	7221	71.00	44.27
22803	SL-16 R/B	7214	70.99	43.29

Table 30: Removal sequence for Test case 1 for a Δv budget of 3 km/s

NORAD ID	Name	Semimajor axis (km)	Inclination (deg)	Criticality
17974	SL-16 R/B	7213	71.01	41.74
23088	SL-16 R/B	7221	71.00	44.27
19120	SL-16 R/B	7206	71.01	41.11
26070	SL-16 R/B	7219	71.00	49.40
16182	SL-16 R/B	7216	71.00	41.97
19650	SL-16 R/B	7218	71.00	42.07
31793	SL-16 R/B	7222	70.98	51.69
23705	SL-16 R/B	7220	71.03	42.51
25407	SL-16 R/B	7218	71.01	42.07

6.4.2 *Test Case 2 (98 degrees of inclination)*

The Test Case 2 involves 30 objects with an inclination around 98 degrees. Table 31 summarizes the results of the four instances solved for this test case. Regarding the instances with refills, the heuristic

initial guess process has been able to determine the global optimality of the instance with one refill and that the obtained criticality of the instance with two refills differs from the global optimum, at most, 0.94%. Regarding the instances without refills, it can be seen that the instance with a capacity of 2km/s is able to remove the same number of objects as both instances with refills. In turn the instance with a capacity of 3km/s is able to remove an additional object. As a result, the instance with two refills shows an even worse case of diminishing returns than the one observed in Test Case 1.

Table 31: Results comparison for Test case 2

Refills	Vehicle capacity (m/s)	Iterations	Removed Objects	Criticality	Optimality gap (%)
2	1000	1	8	184.32	0
3	1000	3	8	188.6	0.94
0	2000	1	8	186.63	0
0	3000	1	9	201.29	0

Tables 32, 33, 34 and 35 show the removal sequences of the four instances. Just like in Test Case 1, it can be seen that the four instances share the majority of the objects. Specifically, the objects, with Norad IDs 37932, 25400, 33272, 27386 (ENVISAT), 41858 and 44548.

Table 32: Removal sequence for Test Case 2 with one refill

NORAD ID	Name	Semimajor axis (km)	Inclination (deg)	Criticality
37932	CZ-2D R/B	7196	98.69	13.77
25400	SL-16 R/B	7185	98.64	41.38
27597	ADEOS 2	7178	98.57	12.14
33272	COSMOS 2441	7098	98.11	23.94
45722	CZ-2C R/B	7116	98.60	10.38
27386	ENVISAT	7143	98.16	50.43
41858	CZ-2D R/B	7150	98.51	17.00
44548	CZ-2D R/B	7139	98.14	15.28

Table 33: Removal sequence for Test Case 2 with two refills

NORAD ID	Name	Semimajor axis (km)	Inclination (deg)	Criticality
25861	SL-16 R/B	7012	98.20	10.92
27601	H-2A R/B	7163	98.19	15.88
44548	CZ-2D R/B	7139	98.14	15.28
41858	CZ-2D R/B	7150	98.51	17.00
27386	ENVISAT	7143	98.16	50.43
33272	COSMOS 2441	7098	98.11	23.94
25400	SL-16 R/B	7185	98.64	41.38
37932	CZ-2D R/B	7196	98.69	13.77

Table 34: Removal sequence for Test case 2 for a Δv budget of 2 km/s

NORAD ID	Name	Semimajor axis (km)	Inclination (deg)	Criticality
25400	SL-16 R/B	7185	98.64	41.38
37932	CZ-2D R/B	7196	98.69	13.77
27597	ADEOS 2	7178	98.57	12.14
33272	COSMOS 2441	7098	98.11	23.94
27386	ENVISAT	7143	98.16	50.43
28480	CZ-2C R/B	7184	98.05	12.69
44548	CZ-2D R/B	7139	98.14	15.28
41858	CZ-2D R/B	7150	98.51	17.00

Table 35: Removal sequence for Test case 2 for a Δv budget of 3 km/s

NORAD ID	Name	Semimajor axis (km)	Inclination (deg)	Criticality
25861	SL-16 R/B	7012	98.20	10.92
27601	H-2A R/B	7163	98.19	15.88
28480	CZ-2C R/B	7184	98.05	12.69
44548	CZ-2D R/B	7139	98.14	15.28
27386	ENVISAT	7143	98.16	50.43
41858	CZ-2D R/B	7150	98.51	17.00
33272	COSMOS 2441	7098	98.11	23.94
37932	CZ-2D R/B	7196	98.69	13.77
25400	SL-16 R/B	7185	98.64	41.38

6.5 CONCLUSIONS

This manuscript analyzes a mission architecture for active debris removal missions comprising a servicing satellite and a fuel station. A two-stage framework has been proposed to solve this problem.

On the one hand, the upper stage explores promising values of the station location and sends them to the lower stage. On the other hand, the lower stage determines the objects to be removed, the removal sequence and the maneuvers of the servicing satellite so that the aggregated criticality of the selected objects is maximized. This is modelled as a Mixed Integer Linear Programming problem and solved using a branch-and-bound method.

In addition to that, a heuristic initial guess process has been devised. This process not only is able to provide an initial guess for the position of the station, but also can potentially determine the global optimal or near-optimal solutions of the problem.

Two practical cases involving debris clusters with inclinations around 71 and 98 degrees, respectively have been analyzed with the proposed framework.

7

CONSTRAINT PROGRAMMING FOR CONSTELLATION SERVICING

7.1 INTRODUCTION

The formation of high-density clusters of man-made spaceborne objects poses a significant risk for the sustainability of future space operations. Specifically, it facilitates the occurrence of a collisional cascading effect that would result in an uncontrollable generation of space debris fragments [65]. This is particularly critical for regions of special operational interest, such as Low Earth Orbit (LEO) or Geostationary Orbit, because it could render them unusable for their future exploitation. Furthermore, even if the space environment does not reach such critical state, a higher object density entails a potential increase in mission cost and disruptions due to a more frequent necessity of collision avoidance activities [53]. So as to stabilise the population of spaceborne objects, it is necessary to actively remove several high-impact pieces of debris per year [71, 75]. Hence, active debris removal missions must rigorously select the objects to be removed so that their impact in the space environment is maximised [8]. However, these missions not only face technical issues, but also challenges of legal and political nature [103]. Thus complicating their practical implementation.

Currently, several initiatives to deploy large constellations in the LEO region are being carried out, such as Starlink [98], OneWeb [84] and Kuiper [68]. It is expected that the operation of such constellations will include the end-of-life deorbiting of its defunct satellites. For instance, the Inter-Agency Space Debris Coordination Committee recommends that the objects that terminate their operational phase within the LEO region should be deorbited or transferred to an orbit with an expected residual orbital lifetime of 25 years or shorter [37]. More recently, the Federal Communications Commission has considered that such timespan should shortened to, at most, an expected lifetime of five years [36]. Nevertheless, the failure of said disposal processes (either because of a premature failure of a satellite or due to unsuccessful deorbiting manoeuvres) poses a threat, not only for the space environment, but also for the constellation performance. This, along with the absence of the legal issues that the general active debris removal missions face, has motivated the assessment of the feasibility of constellation-servicing debris removal missions [25, 47, 70]. In particular, the Sunrise project, funded by the European Space Agency (ESA), intends to identify affordable active debris removal

strategies for large constellations in LEO. Moreover, this project plans to develop the necessary technologies to perform these missions so as to, eventually, provide a competitive service in the international market. As part of Sunrise, ESA commissioned Phase A studies to different consortia, including one comprising D-Orbit SpA and Politecnico di Milano [22, 35, 57]. After the completion of the Phase A studies, the consortium led by Astroscale was chosen to proceed with the next phase of the project [3].

As the objects to be removed in constellation-servicing debris removal missions are not known beforehand, the preliminary design of such missions requires an exhaustive analysis of complex mission configurations, especially when dealing with the coordination of several servicing satellites. Constraint programming [2] is a classical artificial intelligence paradigm, characterised by its flexibility for the modelling of complex problems [89]. Since its inception, it has proven successful for diverse applications [101] such as vehicle routing [93], scheduling [90] and resource allocation [55]. In the field of space operations, constraint-based techniques have been extensively used for mission planning and scheduling [33, 87], involving applications such as Earth observation [49, 88] and deep space exploration missions [59]. More specific applications of constraint-based techniques include NASA's EUROPA planning tool [13], the scientific experiment scheduling of the Rossetta/Philae mission [97] and the mission planning of Orbital Express [67].

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

First, Constraint Programming techniques are used to configure a general framework for preliminary mission analysis. Then, that methodology is particularised for the constellation-servicing debris removal mission considered in a Phase A study, developed by the D-Orbit SpA and Politecnico di Milano consortium under ESA's Sunrise project [57]. In particular, two application cases have been evaluated. Both cases consider a constellation comprising a set of orbital planes with identical inclination, but shifted in Right Ascension of Ascending Node (RAAN). In turn, each of those planes contains a set of satellites that describe an identical circular orbit, but are shifted in

angular position within that orbit. Nevertheless, each of the cases uses a different strategy to remove a set of defunct satellites located within the constellation. On the one hand, the chaser case involves a servicing satellite that rendezvouses with the failed satellites of the constellation and directly transports them to a disposal orbit. On the other hand, the mothership case comprises a servicing satellite that installs deorbiting kits on each of the failed satellites, except for the one removed in the last place which is deorbited by the servicing satellite itself. This way, the servicing satellite will only transport this object, while the deorbiting kits will carry out the disposal of the rest of them.

The remainder of this manuscript is organized as follows. Section 7.2 provides a general description of the proposed Constraint Programming framework. Section 7.3 particularises the proposed methodology for the chaser application case. Section 7.4 does the same for the mothership case. Section 7.5 shows the results that the proposed framework obtains for both application cases. Finally, Section 7.6 summarises the main conclusions of this work.

7.2 METHODOLOGY

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

7.2.1 *Constraint Programming resolution process*

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

provides a great flexibility to develop detailed models of complex problems.

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

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

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

7.2.2 *Constraint Programming for mission analysis*

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

The kind of constraints considered for this application impose a maximum value to a particular performance cost. Hence, the feasibility of a constraint can be evaluated by means of obtaining upper and lower bounds of such performance cost. This way, if the lower bound is greater than the required cost, the constraint is infeasible for the whole variable domain of the evaluated problem instance. This instance is regarded as *rejected*. Likewise, if the upper bound is less that the required cost, the constraint is feasible for the whole domain

of the instance. This problem instance is regarded as *accepted*. However, if the required cost is contained within the bounds, the problem instance is feasible only for some values of the domain. Thus, this instance is considered *inconclusive* and has to be split into smaller ones to accurately determine its feasibility.

Each of the desired bounds can be obtained solving an optimisation problem, dependent on the controlled and uncontrolled variables. This way, the lower bound can be determined by simply finding the values of the controlled and uncontrolled variables that minimise the performance cost. Likewise, selecting the values of the controlled and uncontrolled variables that maximise the performance cost results in an upper bound. However, albeit simple to obtain, this is not the tightest bound, nor the most logical, because a sound mission analysis is supposed to select the most advantageous values of the controlled variables. Therefore, a tighter upper bound is computed when simultaneously using the uncontrolled variables to maximise the cost and the controlled variables to minimise it. Specifically, the former upper bound represents the worst feasible solution, while the latter stands for the worst optimal solution. Thus they are referred as feasibility and optimality upper bounds, respectively.

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

The resolution process that collects the previous concepts is depicted in Figure 16. A problem instance is selected and its feasibility is sequentially evaluated for each of the constraints. If the instance is rejected by one of the constraints, it is not necessary to evaluate its feasibility for the remaining ones and all the more restrictive problem instances are rejected. In turn, if the instance is deemed inconclusive by one of the constraints, it cannot be accepted by the subsequent constraints (i.e., it will remain inconclusive unless a subsequent constraint rejects it). However, if the instance is still considered inconclusive after the last constraint has been evaluated, it is split into simpler instances. Consequently, those new instances are added to the problem instances set. Finally, if the instance is accepted by every constraint, all the less restrictive problem instances are accepted. This process continues until the whole set of problem instances has been evaluated.

7.3 ACTIVE DEBRIS REMOVAL MISSION: CHASER CASE

The chaser case involves the use of a set of servicing satellites, i.e. chasers, to remove defunct satellites within a constellation. In particular, the defunct satellites are directly transported to a disposal orbit

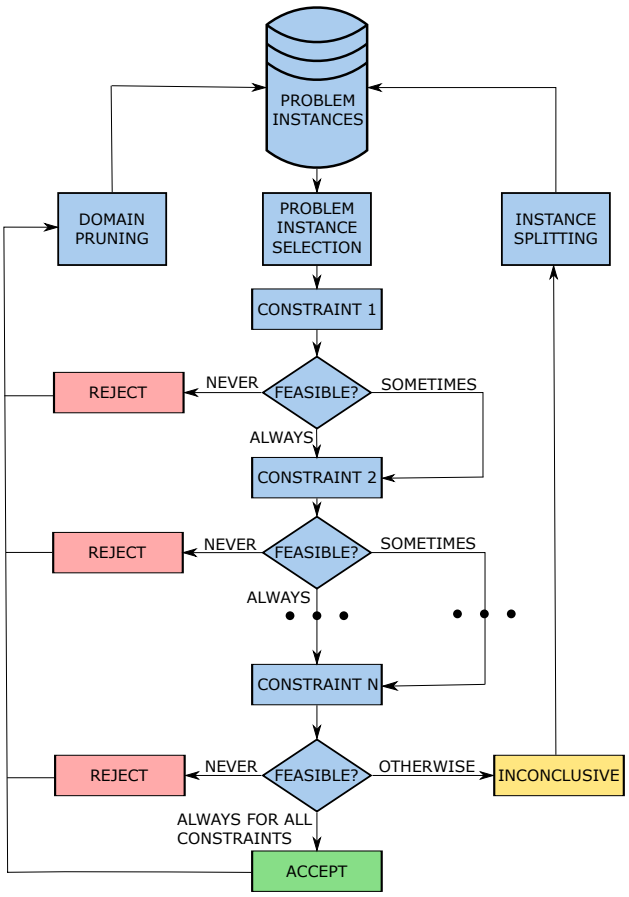


Figure 16: Problem resolution process

by a chaser. This way, each chaser is assigned a set of orbital planes of the constellation and will perform round trips between those planes and their corresponding disposal orbits until the totality of the defunct objects located within those planes is removed. Specifically, each chaser carries out the following sequence of actions:

1. Remaining in the injection orbit until the RAAN of the first constellation plane is achieved. This RAAN variation is exclusively produced by the nodal drift resulting from the J_2 perturbation.
2. Transferring to the first constellation plane and rendezvousing with the first target.
3. Transferring the first target to its corresponding disposal orbit.
4. Repeating Steps 2 and 3 until the first constellation plane is cleared.
5. Transferring to a drifting orbit and remaining there until the RAAN of the subsequent constellation plane is achieved. Just

like in Step 1, this RAAN variation is exclusively produced by the J_2 perturbation.

6. Repeating Steps 2 to 5 until all the constellation planes assigned to the concerning chaser are cleared.

7. The chaser will deorbit together with the last target.

7.3.1 *Predefined mission choices*

The proposed methodology will be used to assess the performance of the choices made in previous mission analyses [35, 57]. In particular, such predefined parameters can be classified into the following categories:

Chaser parameters:

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

Constellation parameters:

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

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

Consequently, the initial problem instances can be designated by a tuple (N, P), where N is the number of objects to be removed and P is

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

7.3.2 *Feasibility bounds*

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

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

The feasibility bounds are directly related with the total RAAN difference traversed by the chaser. Thus, the upper bound is obtained when the last constellation plane to be visited has an initial RAAN very similar to the one of the injection orbit. This way, the chaser has to perform a virtually whole RAAN revolution. In turn, the lower bound is obtained when the RAAN of the injection orbit is identical to the one of the first constellation plane to be visited and the subsequent constellation planes are adjacent to it.

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

7.3.2.1 *Fuel consumption constraint*

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

$$m_i = \left(m_{i+1} + \alpha_i m_{obj}\right) \exp\left(\frac{\Delta V_i}{g_0 I_{SP}}\right) - \alpha_i m_{obj} \tag{157}$$

where i indexes the set of performed manoeuvres (in chronological order), m_i is the mass of the chaser after manoeuvre i, m_{obj} is the mass of a defunct satellite, α_i is 1 if the chaser is transporting a defunct satellite during manoeuvre i (is 0 otherwise), ΔV_i is the ΔV spent during manoeuvre i, g_0 is the gravity acceleration at the Earth's surface and I_{SP} is the specific impulse of the chaser's thruster. This way, given the whole set of ΔV_i values and starting with a m_{i+1} equal to the dry mass of the chaser, Eq. (157) can be iterated to obtain the initial wet mass of the chaser.

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

$$\Delta V = \sqrt{V_1^2 + V_2^2 - 2V_1 V_2 \cos\left(\Delta i\right)} \tag{158}$$

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

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

7.3.2.2 *Mission time constraint*

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

$$\begin{aligned} \Delta t = {} & \Delta t_{Tdf} + (N - P + 1)\Delta t_{DC} + (P - 1)\Delta t_{DN} + \\ & + \Delta t_{RI} + (N - P)\Delta t_{RC} + (P - 1)\Delta t_{RN} \end{aligned} \tag{159}$$

where Δt_{Tdf} is the aggregated time spent while coasting in the different drifting orbits, Δt_{RI} is the time elapsed between the departure from the initial injection orbit and the rendezvous with the first object, and Δt_{DC} and Δt_{RC} are, respectively, the time spent to transport an object to its corresponding disposal orbit and the time elapsed between departing such disposal orbit and the rendezvous with the

subsequent object; both terms consider that such object is situated in the same constellation plane as the previously removed one. In turn, Δt_{DN} and Δt_{RN} are analogous to the previous terms, but for the case in which the subsequent object is noncoplanar with the formerly removed one.

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

$$\Delta t_{df} = \frac{\Delta\Omega + 2\pi C}{\dot{\Omega}_2 - \dot{\Omega}_{df}} \tag{160}$$

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

$$\dot{\Omega} = -\frac{3nR_{\oplus}^2 J_2}{2p^2}\cos(i) \tag{161}$$

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

Aside from the coasting intervals, the rest of the manoeuvres can be generalised with a two-stage strategy involving a phasing stage and a Hohmann-like transfer with inclination change (although it is not necessary for all of them to comprise a phasing stage or an inclination change). Thus, the remaining terms of Eq. (159) can be modelled as:

$$\Delta\hat{t} = KT_{pha} + \frac{1}{2}T_{tra} \tag{162}$$

where $\Delta\hat{t}$ represents a term of Eq. (159) (excluding Δt_{Tdf}), K is the number of revolutions performed by the chaser during the phasing stage and T_{pha} and T_{tra} are the orbital periods of the phasing and transfer orbits, respectively. Moreover, K and T_{pha} have to be chosen so that the mean motion difference between the target and phasing orbits produces a desired difference in true anomaly in an integer number of revolutions:

$$\Delta\theta + 2\pi C + (n_2 - n_{pha})KT_{pha} = 0 \tag{163}$$

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

Reformulating Eq.(163) in terms of the orbital periods and isolating K gives:

$$K = \frac{\Delta\theta + 2\pi C}{2\pi\left(1 - \frac{T_{pha}}{T_2}\right)} \tag{164}$$

Three unknown variables of Eq.(164), namely, K, C and T_{pha}, have to be determined. An initial step to configure an efficient algorithm to solve this equation is to analyse the domain of the concerned variables.

Regarding T_{pha}, in Section 7.3.2.1, it was stated that the positions of the objects within their constellation plane do not affect the ΔV consumption. It is due to the fact that the phasing stage does not produce a net ΔV consumption, i.e., the ΔV spent to reach the phasing orbit and, subsequently, the transfer orbit is identical to the ΔV that would be spent to directly reach the latter. This entails that the semimajor axis of the phasing orbit must have a value located between the ones corresponding to the semimajor axes of the initial orbit of the current manoeuvre and the transfer orbit. In terms of orbital periods, it means that $T_{pha} \in [\min(T_1, T_{tra}), \max(T_1, T_{tra})]$, where T_1 is the period of the initial orbit of the current manoeuvre.

Regarding K and C, in order to minimise KT_{pha}, K has to be the smallest natural number possible. In addition, for K to be positive, C must have the same sign as $(1 - T_{pha}/T_2)$ and its absolute value has to be as small as possible to minimise K. Thus, a lower bound of $|C|$ can be obtained when substituting $K = 1$ as well as the minimum and maximum values of T_{pha} in Eq.(164) and selecting the minimum $|C|$ from the two values obtained. Then, as C must be an integer, if C has to be positive, it is rounded up, otherwise, it is rounded down.

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

1. The lower bound of C is substituted in Eq. (164).
2. The minimum and maximum values of T_{pha} are introduced in that equation, resulting in the extremes of an interval of possible values of K.
3. If there are natural numbers within such interval, K is the smallest of them.
4. Otherwise, increase $|C|$ in 1 and repeat the previous two steps until K is determined.
5. Once C and K are known, they are substituted in Eq.(164) to obtain T_{pha}.

7.3.3 *Optimality upper bounds*

The feasibility upper bounds are obtained when both the controlled and uncontrolled variables are selected to produce the least advantageous value of a particular constraint. However, in practice, the controlled variables will be carefully selected so as to produce the most advantageous results for the mission. The optimality upper bounds take into account this concept by means of the interaction of two antagonistic optimisation processes. Specifically, the uncontrolled variables are chosen to worsen the mission performance, while the controlled ones are selected to improve the constraint value. As a result, the optimality bounds represent the tightest bounds that can be obtained for a particular constraint value, but at a greater computational complexity.

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

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

Strictly speaking, the quantity added to the initial position of the chaser to connect the different solutions is not identical to the constant added to the position of the objects. Instead, such constant has to be added to the position that the chaser would achieve within the constellation plane after the drifting and the necessary transfers, but without performing any phasing manoeuvres. Then, the transfers and drift would have to be propagated backwards to achieve the initial position of the chaser. However, this process can be circumvented by directly considering such chaser position within the constellation plane during the problem resolution. As a consequence, the aggregated RAAN traversed by the chaser while drifting can be uncloupled from this problem. Thus obtaining the optimality upper bound by simply considering the largest possible traversed RAAN.

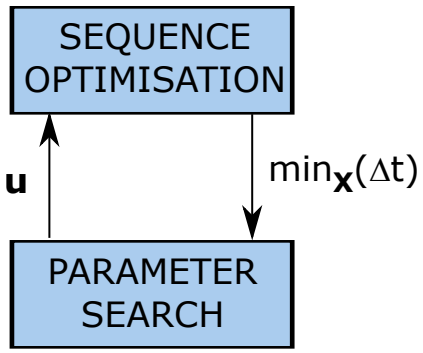


Figure 17: Optimality upper bound process

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

$$\max_{\mathbf{u}} \{\min_{\mathbf{X}} \{\Delta t(\mathbf{u}, \mathbf{X})\}\} \tag{165}$$

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

The resolution process of this problem follows the structure shown in Figure 17. The upper lever, hereinafter referred to as parameter search, explores $\mathbf{u}$ and sends promising values to the lower lever. In turn, the lower level obtains the optimal removal sequence for the received $\mathbf{u}$ and provides the resulting Δt to the upper level. This way, the upper level can use the Δt information to select a subsequent $\mathbf{u}$ that can potentially produce a worse Δt.

7.3.3.1 *Parameter search*

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

$$u_i \leqslant u_{i+1} \quad \forall i : 2 \leqslant i \leqslant N \tag{166}$$

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

Therefore, considering the sequence optimisation problem as a black-box function, the parameter search problem involves the maximisation of the objective function defined by Eq. (165), subject to Eq. (166).

As Eq. (165) is non-smooth and discontinuous, derivative-free methodologies are proposed to solve this problem. In addition, said function is computationally expensive to evaluate because it requires the resolution of an Integer Programming problem. This arises a dilemma about the trade-off between objective function evaluations and solution quality. As a result, two different methodologies are proposed to solve the parameter search problem.

On the one hand, Generalised Pattern Search [72] constitutes a derivative-free direct search methodology that follows a similar strategy to steepest descent approaches. This way, a single solution is iteratively improved by means of sampling its neighbouring points. Thus being prone to the obtention of disadvantageous local optimal solutions and, consequently, being sensitive to the initial guess used to initialise the search. On the other hand, Evolutionary Algorithms provide a better search space exploration by means of maintaining a population of possible solutions, but at the cost of a considerably larger number of objective function evaluations.

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

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

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

7.3.3.2 *Sequence optimisation*

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

problem. In particular, this problem intends to minimise the total Δt spent during the mission:

$$\min \left\{ \sum_{i \in D} \sum_{\substack{j \in D \\ i \neq j}} \Delta t_{ij} X_{ij} \right\} \tag{167}$$

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

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

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

$$X_{ii} = 0 \quad \forall i \in D \tag{168}$$

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

$$\sum_{\substack{j \in D \\ i \neq j}} X_{ij} = Y_i \quad \forall i \in D \tag{169}$$

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

Likewise, each removed object has exactly one incoming transfer:

$$\sum_{\substack{j \in D \\ i \neq j}} X_{ji} = Y_i \quad \forall i \in D \tag{170}$$

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

$$\sum_{i \in D} Y_i = N + 1 \quad \forall i \in D \tag{171}$$

Note that if the whole set of objects has to be removed, case known as Travelling Salesman Problem (TSP) [39], this latter constraint is equivalent to imposing that $Y_i = 1 \ \forall i \in D$.

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

A common strategy to eliminate potential subtours entails the consideration of additional ad-hoc constraints of the following form [39]:

$$\sum_{i \in R} \sum_{\substack{j \in R \\ i \neq j}} X_{ij} \leqslant |R| - 1 \tag{172}$$

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

On the one hand, for problems involving five or less objects, the addition of Constraints (172) particularised for two-object subtours (i.e. round trips between two objects) is enough to prevent the appearance of subtours. Such constraints can be defined as follows:

$$X_{ij} + X_{ji} \leqslant 1 \quad \forall i \in D \quad \forall j \in D : (i < j) \tag{173}$$

On the other hand, for larger problems, Constraints (172) can be dynamically generated during the resolution of the problem. This way, if a solution with subtours is found, the solution will be rejected and the corresponding subtour elimination constraints will be added to the model. This strategy is usually very efficient, but its performance significantly degrades for problems with a large number of relevant subtours, specially for instances involving objects distributed within different orbital planes.

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

Such formulations have been thoroughly explored for the Travelling Salesman Problem [69]. The most notable of them [81] involves the introduction of additional variables z_i that directly determine the order in which each node is visited. Assuming that $z_1 = 0$, i.e., the

chaser starts at the node 1, the domain of the rest of those variables is $z_i \in [1, |D| - 1]$ and their corresponding order can be obtained with the following set of constraints:

$$z_i - z_j + |D| X_{ij} \leqslant |D| - 1 \quad \forall i \in D : (i > 1), \quad \forall j \in D : ((j > 1) \wedge (i \neq j)) \tag{174}$$

The TSP no-subtour formulations achieve a weaker linear relaxation than the one that would be obtained with the use of Constraints (172) or require a large number of constraints [69]. Thus, the dynamic elimination is often preferred. However, they can be useful when there is a large number of relevant subtours.

A more general no-subtour formulation, suitable for problems in which it is not necessary to remove all the candidate objects, was proposed in [8]. It includes the sequence information within the X_{ij} variables, transforming them into the new X_{ijk} variables and substituting each X_{ij} instance for the expression $\sum_{k \in K} X_{ijk}$. This way, X_{ijk} is 1 if and only if the k-th transfer is performed between objects i and j. Moreover, one transfer corresponds to each position k:

$$\sum_{i \in D} \sum_{\substack{j \in D \\ i \neq j}} X_{ijk} = 1 \qquad \forall k \in [1, N + 1] \tag{175}$$

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

$$\sum_{j \in D} X_{1j1} = 1; \qquad \sum_{j \in D} X_{j1(N+1)} = 1 \tag{176}$$

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

$$\sum_{\substack{i \in D \\ i \neq j}} X_{ijk} = \sum_{\substack{i \in D \\ i \neq j}} X_{ji(k+1)} \quad \forall j \in D, \quad \forall k \in [1, N] \tag{177}$$

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

Table 36 summarises the recommended formulation selection for different kinds of problem instances. Note that it is not clear which

formulation to use for cases that involve a large set of candidate objects, distributed among different orbital planes, such that a large subset of those objects is required to be removed. The reason for that is the degradation of the performance of both the dynamic elimination strategy and the general no-subtour formulation for such kind of instances.

Table 36: MIP formulation selection criteria

$\|D\|$	Object distribution	N	Formulation
$\leqslant 5$	Any	$[1, \|D\|]$	Two-object constraints [39]
Small	Any	$[1, \|D\|]$	Dynamic elimination [39]
Large	Coplanar	$\|D\|$	Dynamic elimination [39]
Large	Noncoplanar	$\|D\|$	TSP No-subtour [69, 81]
Large	Any	$\ll \|D\|$	General No-subtour [8]
Large	Noncoplanar	$\approx \|D\|$	Undetermined

7.4 ACTIVE DEBRIS REMOVAL MISSION: MOTHERSHIP CASE

The mothership case, just like the chaser case, involves the use of a set of servicing satellites, i.e. motherships, to remove defunct satellites within a constellation. However, in this case, the defunct satellites are transported to their corresponding disposal orbits by deorbiting kits, which have been previously attached to them by a mothership. This way, each mothership rendezvous with each of its assigned objects to deploy the deorbiting kits and will only transfer to a disposal orbit when performing the removal of its last associated object. During that manoeuvre, the mothership will transport such object to the disposal orbit so that both can simultaneously reenter the atmosphere. Hence, the sequence of actions carried out by each mothership is analogous to the one explained in Section 7.3, save for the intermediate transfers to the disposal orbit.

7.4.1 *Predefined mission choices*

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

Mothership parameters:

- Maximum wet and dry mass of each mothership.
- Mass of the deorbiting kits.
- Semimajor axis, inclination and eccentricity of the initial injection orbit.

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

Constellation parameters:

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

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

7.4.2 *Feasibility bounds*

The requirements imposed by the previous mission analysis involve limitations in the maximum mission time and the ΔV consumed during the mission. Therefore, the most and least advantageous values of the object distribution within the different constellation planes and the initial positions of the different objects have to be determined.

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

instantaneously transferring back to the constellation orbit, i.e., the phasing is directly achieved by the transfer orbit.

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

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

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

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

$$\min \left\{ \frac{2\pi}{N'_1} + \sum_{\substack{p \in P \\ p>1}} \frac{\pi}{N'_p} \right\} \tag{178}$$

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

$$\sum_{p \in P} N'_p = N \tag{179}$$

Moreover, N'_p must be natural numbers. However, when relaxing such integrality condition, the problem defined by Eqs. (178,179) has the following analytical solution:

$$N'_1 = \frac{N\sqrt{2}}{P - 1 + \sqrt{2}}; \qquad N'_p = \frac{N}{P - 1 + \sqrt{2}} \quad \forall p \in P : p > 1 \tag{180}$$

Then, two possible integer solutions can be obtained, resulting from rounding up or down the value of N'_1. The rest of variables can be rounded so that Eq. (179) is fulfilled and both solutions are computed to select the one that minimises Eq. (178).

7.4.2.1 ΔV *constraint*

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

$$\Delta V = \Delta V_{RI} + (N - P)\Delta V_{RC} + (P - 1)\Delta V_{RN} + \Delta V_{D} \tag{181}$$

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

Each of the terms of the right hand side of Eq. (181) comprises several impulses (modelled using Eq. (158)) arranged to configure Hohmann-like transfers with inclination changes as well as intermediate phasing and drifting orbits. The semimajor axis and inclination changes performed during each of those impulses are optimised so that their corresponding term is minimised, while complying with the predefined mission choices.

It has to be noted that the predefined mission choices, along with the optimised terms of the right hand side of Eq. (181), unambiguously define the spent ΔV for each (N, P) tuple. Hence, both feasibility bounds associated to the ΔV constraint collapse into a single quantity regardless of the values of the uncontrolled variables.

7.4.2.2 *Mission time constraint*

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

$$\Delta t = \Delta t_{Tdf} + \Delta t_{RI} + (N - P)\Delta t_{RC} + (P - 1)\Delta t_{RN} \tag{182}$$

where Δt_{Tdf} is the aggregated time spent while coasting in the different drifting orbits and the subindices RI, RC and RN are analogous to the ones found in Eq. (181), but applied to the transfer time. Each of the individual drifting times included in Δt_{Tdf} can be obtained using Eq. (160). In turn, the rest of the Δt components of Eq. (182) can be obtained by means of the resolution of Eq. (162).

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

7.4.3 *Optimality bounds*

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

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

7.4.3.1 *Phasing and drifting orbit optimisation*

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

$$\min \left\{ \sum_{k \in K} \Delta t_{pha}\left(a_k\right) + \sum_{\ell \in L} \Delta t_{df}\left(i_\ell\right) \right\} \tag{183}$$

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

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

$$\sum_{k \in K} \Delta V_{pha}\left(a_k\right) + \sum_{\ell \in L} \Delta V_{df}\left(i_\ell\right) = \Delta V^* \tag{184}$$

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

This problem can be readily solved with conventional Nonlinear Programming techniques. However, the dual-based methodology proposed in [10] provides an efficient way to obtain the global optimum of the problem. For the sake of completeness, the application of such methodology to the problem at hand is subsequently explained in a concise manner.

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

$$\mathcal{L} = \sum_{k\in K} \Delta t_{pha}(a_k) + \sum_{\ell\in L} \Delta t_{df}(i_\ell) - \lambda \left(\sum_{k\in K} \Delta V_{pha}(a_k) + \sum_{\ell\in L} \Delta V_{df}(i_\ell) - \Delta V^* \right) \tag{185}$$

where λ is the dual variable associated to Eq. (184).

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

$$\frac{d\Delta t_{pha}}{da_k}(a_k) - \lambda \frac{d\Delta V_{pha}}{da_k}(a_k) = 0 \quad \forall k \in K \tag{186a}$$

$$\frac{d\Delta t_{df}}{di_\ell}(i_\ell) - \lambda \frac{d\Delta V_{df}}{di_\ell}(i_\ell) = 0 \quad \forall \ell \in L \tag{186b}$$

The derivative chain rule can be used to isolate λ in Eqs. (186), resulting in the following conservation law:

$$\lambda = \frac{d\Delta t_{pha}}{d\Delta V_{pha}}(a_k) = \frac{d\Delta t_{df}}{d\Delta V_{df}}(i_\ell) \qquad \forall k \in K, \quad \forall \ell \in L \tag{187}$$

Eqs. (187) can be inverted and substituted into Eq. (184) to configure the following univariate function:

$$\phi(\lambda) = \sum_{k\in K} \Delta V_{pha}(a_k(\lambda)) + \sum_{\ell\in L} \Delta V_{df}(i_\ell(\lambda)) - \Delta V^* \tag{188}$$

A root of Eq. (188) automatically fulfills the optimality conditions, i.e., Eqs. (184,187). Hence, the solution of the problem simply involves the determination of λ, regardless of the number of phasing and drifting orbits. Then, the values of a_k and i_ℓ can be retrieved with the inverse of Eqs. (187).

7.4.3.2 *Phasing orbit correction*

The previous problem regards a_k as a continuous variable. However, only the values that result in an integer number of revolutions of the

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

$$\min \sum_{k \in K} \Delta(\Delta t_k) \Psi_k \tag{189}$$

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

$$\sum_{k \in K} \Delta(\Delta V_k) \Psi_k \leqslant \Delta V' \tag{190}$$

where $\Delta(\Delta V_k)$ is the ΔV difference between the rounded down and the rounded up revolutions and $\Delta V'$ is the ΔV surplus to redistribute. Eqs. (189,190) define an Integer Linear Programming problem that can be readily solved with branch-and-bound methods.

7.5 RESULTS

The mission analysis from Ref. [57] is subsequently compared with the results obtained with the proposed methodology. Specifically, predefined performance baselines are defined for both application cases and the achieved level of fulfilment of such baselines is discussed. Table 37 shows the predefined mission parameters shared by both application cases.

7.5.1 *Chaser case*

Table 38 shows the requirements imposed for the chaser case. In addition, its associated performance baseline involves removing three objects located in different orbital planes. Moreover, it is assumed that the chaser will be deployed with a rideshare launch. Thus, the initial RAAN associated to the injection orbit is considered as an uncontrolled variable. This implies the analysis of two possible mission geometries. On the one hand, the case with *Type 1 Positioning* considers that such initial RAAN is not included within the interval defined by the constellation planes to be cleared. On the other hand, the *Type*

Table 37: Predefined parameters for both application cases

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

2 *Positioning* involves an initial RAAN inside that interval, resulting in a more disadvantageous case.

Table 38: Mission requirements for the chaser case

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

Tables 39 and 40 summarise the mission analysis carried out in Ref. [57]. In particular, the inclination of the drifting orbits depend on the initial positioning of the chaser. In turn, the disposal orbit associated to an object depends on the constellation plane in which the next object to be removed is located, as well as the initial positioning of the chaser.

Table 39: Drifting orbit parameters for the chaser case

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

Consequently, according to [57], the aforementioned parameter selection makes it possible for the chaser to remove three coplanar objects within the required mission time, regardless of the initial positioning of the chaser. However, in order to remove three noncoplanar

Table 40: Disposal orbit parameters for the chaser case

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

objects, it would be necessary to increase the wet mass of the chaser by 5 and 35 kg for the initial positioning of Type 1 and 2, respectively.

Hence, the results of the proposed methodology are compared with such conclusions. The first step in its application is the configuration of the problem instances. Specifically, as the (N, P) combinations are going to be evaluated, an upper bound of the maximum N to be evaluated has to be obtained. Such maximum number of objects can be readily computed by iterating backwards Eq. (157), starting from the dry mass of the chaser, until a mass of the chaser greater than the maximum wet mass is achieved. This yields that, under the imposed requirements, it is impossible to remove more than three objects with a single mission.

Consequently, Figure 18 shows the results of computing the feasibility bounds for the (N, P) problem instances of up to three objects, where the green instances are feasible in any case, the red ones are always infeasible and the yellow ones are inconclusive. The evaluated instances are highlighted with black squares, while the feasibility of the rest was concluded with constraint propagation. So far, the feasibility showed in Figure 18 is in line with the conclusions of [57]. That is, for Type 2 positioning, it is only possible to remove three objects if they are coplanar. In turn, for Type 1 positioning, there are cases in which it is not feasible to remove three noncoplanar objects. However, the mission time upper feasibility bound of the $(3, 3)$ instance, for Type 1 positioning, violates the maximum mission time by just 2.8 hours. Moreover, further splitting such problem instance, it is determined that it is feasible for any aggregated $\Delta\Omega$ traversed by the chaser with a value lower than 359.961 degrees. Thus, unlike [57] proposes, it is not worth to modify the design of the chaser for such a small and improbable violation of the maximum mission time.

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

As the considered sequence involves at most four objects (three objects to be removed plus the dummy object), the formulation with

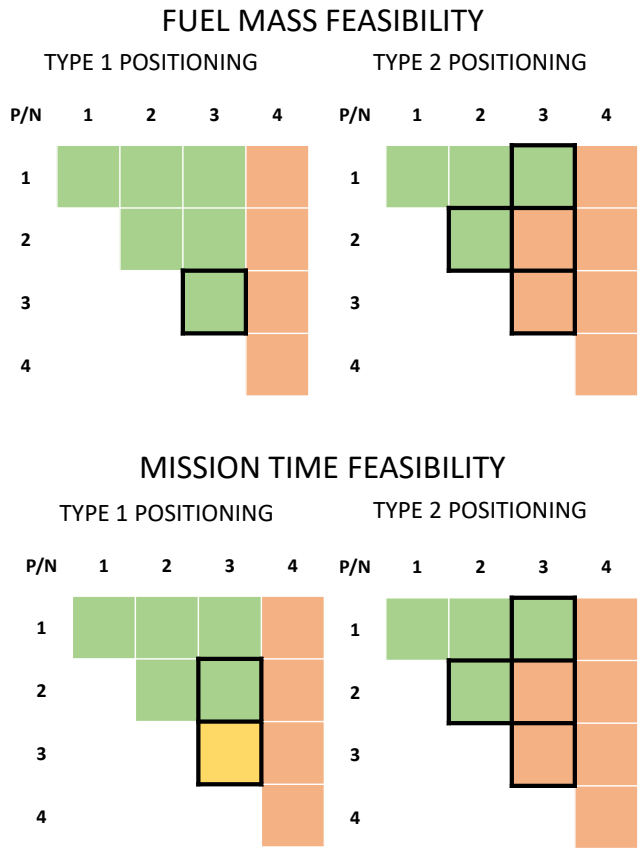


Figure 18: (N, P) instances diagram for the chaser case

two-object constraints is used for the sequence optimisation level. Regarding the parameter search level, it is solved using the two methodologies mentioned in Section 7.3.3.1 (Generalised Pattern Search and an Evolutionary Algorithm), as well as the two initial guesses discussed there (i.e., the object distribution that gives worst feasible rendezvous time and all the objects with an identical position).

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

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

Table 41 shows the Δt ratio between the worst aggregated value for the optimal rendezvous time (corresponding to the optimality upper bound) and the worst aggregated value for the feasible rendezvous time (corresponding to the feasibility upper bound). It has to be noted that the problem can be decomposed into each individual constellation plane. Thus, the depicted solutions represent the removal of two or three objects within a single plane, for the cases in which it is the first plane cleared or one of the subsequent ones.

Table 41: Worst rendezvous time ratio for plane-decomposed cases

Number of objects	Orbital plane	Optimisation technique	Initial guess	Number of iterations	Function evaluations	Δt_{RC} ratio
2	First	P. Search	Identical	26	147	0.9051
2	First	P. Search	Worst	60	270	0.9051
2	First	Genetic	Identical	51	2450	0.9051
2	First	Genetic	Worst	51	2450	0.9051
2	Rest	P. Search	Identical	24	129	0.9247
2	Rest	P. Search	Worst	52	178	0.8138
2	Rest	Genetic	Identical	51	2450	0.9247
2	Rest	Genetic	Worst	51	2450	0.9247
3	First	P. Search	Identical	32	301	0.8798
3	First	P. Search	Worst	50	242	0.7156
3	First	Genetic	Identical	51	2450	0.8798
3	First	Genetic	Worst	68	3249	0.7986
3	Rest	P. Search	Identical	28	250	0.8967
3	Rest	P. Search	Worst	40	214	0.3714
3	Rest	Genetic	Identical	51	2450	0.8967
3	Rest	Genetic	Worst	79	3766	0.8715

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

Table 42 shows the aggregated rendezvous time ratio for the multi-plane problem instances, obtained by assembling the plane-decomposed solutions. Obviously, when there is a single object per plane, the feasibility and optimality upper bounds are identical and so are the worst rendezvous times associated to each of those bounds. Moreover, the larger the number of objects within a single plane, the smaller the computed ratio and, thus, more important the influence of the sequence selection in the optimality bounds.

As previously stated, this upper bound improvement is not necessary to accurately determine the feasibility of the considered problem instances, mainly because the majority of the mission time is spent during the drifting phases. However, this could have a great impact for practical cases in which a larger number of objects is removed from a single constellation plane and the chaser is directly injected into it. Furthermore, such larger number of objects entails a more complex optimisation problem, therefore emphasising the importance of a good initial guess like the one proposed in this work.

Table 42: Worst rendezvous time ratio for multi-plane cases

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

7.5.2 *Mothership case*

Table 43 shows the requirements imposed to the mothership case. Its associated performance baseline involves two different scenarios. On the one hand, *Scenario 1* considers nine objects to remove within each of the constellation planes. Each of those planes has associated its own mothership, resulting in a mission involving twelve servicing satellites. It is assumed that the whole mothership set is launched into a single injection orbit with the RAAN of the first constellation plane to be cleared. This way, the remaining motherships will coast in the injection orbit until achieving the RAAN of their associated orbit. Then, they will perform a transfer to rendezvous with one of the objects and, after that, phasing manoeuvres will be carried out to remove the remaining ones. Such phasing manoeuvres involve a transfer to a circular orbit with an altitude of 1195 km, so as to keep a safety distance of 5 km from potential active satellites while coasting in the phasing orbit.

On the other hand, *Scenario 2* also involves removing nine objects with each mothership. However, in this case, those objects are distributed among two adjacent constellation planes. As a result, this

scenario only requires six servicing satellites. After clearing its first associated constellation plane, each mothership will use a drifting orbit to transfer to the subsequent one. In particular, the selected drifting orbit is identical to the constellation orbit save for its inclination, which will have a value of 87.67 deg.

Table 43: Mission requirements for the mothership case

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

The aforementioned mission analysis [57] states that, for Scenario 1, every mothership is able to clear its assigned plane while fulfilling the maximum mission time and ΔV constraints. However, for Scenario 2, if the ΔV constraint is fulfilled, the upper bound for the mission time ranges from 2.2 years (for the first mothership) to 3.4 years (for the sixth mothership).

The results of the proposed methodology are compared with such conclusions. Figure 19 shows the results of computing the feasibility bounds for the (N, P) problem instances of up to nine objects. In particular, this figure depicts the problem instance diagrams for the motherships that serve the planes closer to and farther from the injection orbit, respectively labeled as minimum and maximum drift cases. It has to be noted that the feasibility of both mission time and ΔV requirements is considered in such diagrams, i.e., if one of those constraints is infeasible for an instance, such instance is deemed infeasible. In turn, for an instance to be feasible, both requirements have to be feasible for the whole variable domain. The feasibility showed in Figure 19 is in line with the conclusions of [57]. However, it has been observed that the infeasibility of removing objects dristributed into two planes is due to violations of the maximum mission time constraint. In contrast, the predefined mission parameters result in an unused ΔV of 214.9 $m{\cdot}s^{-1}$. Hence, instead of exploring the particular cases in which the removal of objects distributed into two planes is feasible, it would be of great interest to use the techniques explained in Section 7.4.3 to optimally redistribute such ΔV surplus, thus minimising the mission time.

Figure 20 depicts the inverse of the summands of Eq. (188), i.e., λ as a function of ΔV for each of the problem variables. It has to be noted that, for a particular constellation plane, the conditions of all of the phasing maneuvers (save for the first one) are identical. Hence, the ΔV assigned to the phasing of four of the five objects within the first constellation plane is represented by the solid blue line. Likewise, the phasing of three of the four objects within the second plane is characterised by the solid red line. Furthermore, the value of λ cor-

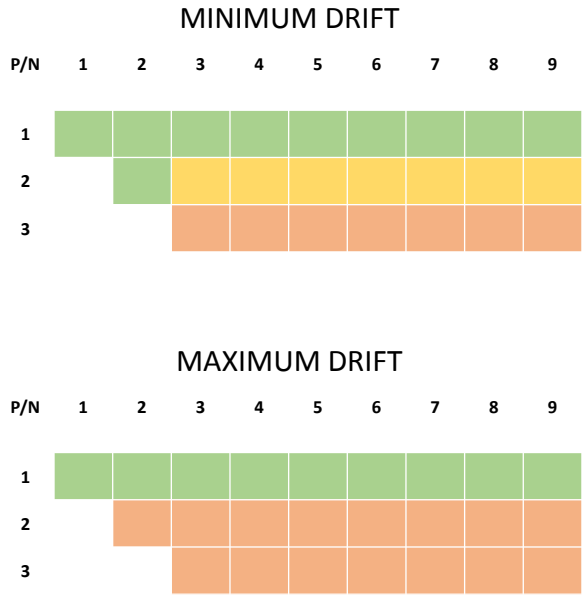


Figure 19: (N, P) instances diagram for the mothership case

responding to a root of Eq. (188) is also portrayed. Consequently, the ΔV assigned to each of the manoeuvres corresponds to the intersection of its corresponding function with the λ value. As the depicted functions show a monotonically increasing behaviour, such intersection is unique and the obtained solution is a global optimum.

Table 44 provides a comparative between the initial inclinations of the injection and drifting orbits, provided by [57], and the values resulting from the computed λ. It also depicts the ΔV allocated to each of those orbits. As the drifting phases have a much greater influence in the mission time than the phasing manoeuvres, the bulk of the ΔV surplus is allocated to increase the nodal drift of the injection and drifting orbits.

Table 44: Optimised drifting orbit inclinations

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

Table 45 shows the semimajor axes associated to each of the phasing orbits, obtained from the computation of λ, as well as their allocated ΔV. Specifically, phasing orbits 1 to 4 are the ones used to rendezvous with objects 2 to 5 within the first constellation plane.

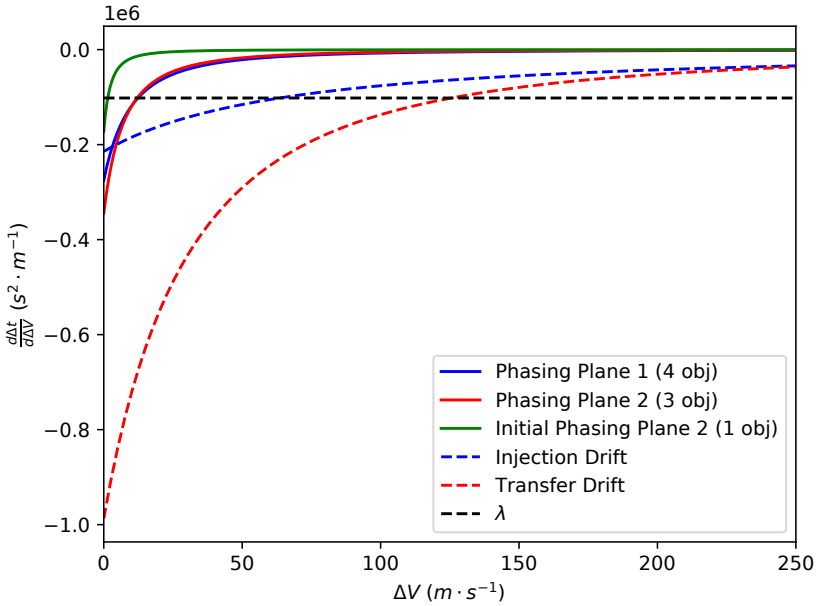


Figure 20: Mothership case parameter correction

Likewise, phasing orbits 5 to 8 are the ones used to rendezvous with the four objects situated within the second plane. Furthermore, a Corr. and ΔV Corr. stand for the corrections that have to be added to the previous columns so that an integer number of revolutions during the phasing manoeuvres is achieved. Those corrections happen to be considerably small. In addition, Δt Dif. represents the difference in mission time produced by them. The aggregated value of such time difference amounts to 3919 seconds, which is negligible with respect to the maximum mission time. Therefore, the solution obtained in the phasing and drifting orbit optimisation can be considered as a good approximation of the mission time, regardless of the integrality of the revolutions of the phasing orbits.

Finally, Figure 21 depicts a comparison between the initial bounds of the mission time and the bounds resulting from the optimised mission, for the problem instance involving nine objects distributed within two constellation planes. The optimised mission shows a remarkable improvement of the mission time, hence fulfilling the maximum mission time constraint with the six servicing satellites and accomplishing the predefined performance baseline for Scenario 2.

7.6 CONCLUSIONS

This manuscript proposes a Constraint Programming framework for the preliminary analysis of space missions. Specifically, it is able to quantify the performance of a set of predefined mission choices with

Table 45: Optimised phasing orbits

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

respect to the mission requirements. Moreover, if a poor performance is shown or if the mission choices have not been previously obtained, appropriate mission choices will be generated so that the desired performance is optimised.

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

The feasibility of an instance depends on a set of controlled and uncontrolled variables and it is determined by means of bounding the range of constraint values that would be obtained for that set of variables. It has to be noted that such bounds are not unique. Hence, two different sets of bounds have been proposed. On the one hand, the feasibility bounds are obtained when using both the controlled and uncontrolled variables to minimise (or maximise) the constraint value. On the other hand, the optimality upper bound is obtained when using the controlled variables to minimise the constraint value, while the uncontrolled variables try to maximise it. Consequently, the optimality bounds provide a tighter interval of constraint values, but at the cost of a greater computational complexity.

The proposed methodology has been particularised for two application cases involving constellation-servicing active debris removal missions, namely, a chaser case and a mothership case. The chaser case involves constraints in the fuel consumption and the mission time. Their corresponding feasibility bounds can be readily computed by an algebraic expression. However, the obtention of the optimality

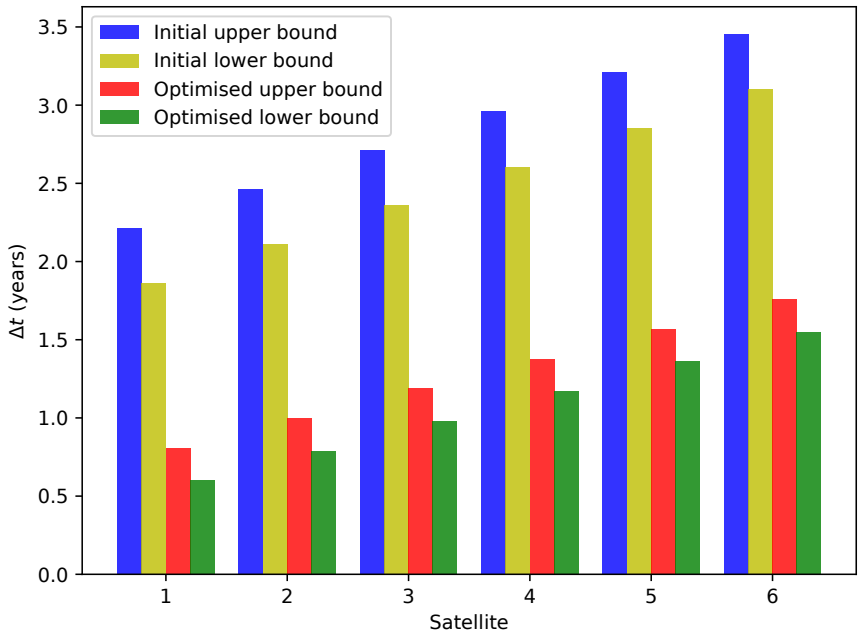


Figure 21: Mothership case mission time comparison. The optimised cases used the surplus ΔV budget to improve the mission time bounds.

upper bound requires the resolution of a Bilevel Mixed Integer Nonlinear Programming problem. In such problem, the initial positions of the objects and the removal sequence are simultaneously chosen so as to respectively maximise and minimise the mission time. The upper level is solved using a derivative-free method. Specifically, a Genetic Algorithm and Generalised Pattern Search have been used. Furthermore, an initial guess that provides, under particular circumstances, a near optimal or even the global optimal solution has been figured out. The lower level is modelled as an Integer Linear Programming problem and solved using Branch-and-Bound techniques.

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

This methodology has been used to evaluate a preliminary mission analysis of both application cases, developed under ESA's Sunrise project. Regarding the chaser case, it has been determined that its associated mission choices achieve a better performance than the one computed in the preliminary analysis. That is, a more precise knowl-

edge about the performance of the preliminary analysis has been gained, which is an effect of the thorough exploration of the search space performed by the proposed methodology. Regarding the mothership case, it has been shown that the preliminary analysis provides a poor performance. As a result, new values for the semimajor axes of the phasing orbits and the inclinations of the injection and drifting orbits have been computed. Thus obtaining significant performance improvements.

Part IV

CONCLUSIONS

All such problems can be formulated as mathematical programming problems. Naturally, we can propose many sophisticated algorithms and a theory but the final test of a theory is its capacity to solve the problems which originated it.

— George Dantzig, Linear Programming and Extensions, 1963

8

CONCLUSIONS AND FUTURE WORK

8.1 OVERALL CONCLUSIONS

To sum up, this thesis focuses on the resolution of optimization problems involving the preliminary analysis of large-scale active debris removal missions. Specifically, the techniques hereby devised to efficiently solve such problems have dealt with three particular challenges. Namely, the combinatorial complexity of the problems, the computational load of the maneuver optimization and the interaction between the combinatorial decisions and the orbital dynamics.

The combinatorial complexity of the problems has been mainly addressed with the MILP no-subtour formulation presented in Chapter 4. This formulation has shown a remarkable performance when dealing with for problem instances involving a large number of objects with diverse orbital configurations situated in the LEO region and has been subsequently used in the application cases discussed in Chapters 5 and 6. However, this formulation is not universally suitable for every application case. Specifically, Chapter 7 discusses the use of different MILP formulations for object pools of different magnitudes and object distributions.

Regarding the maneuver optimization, Chapter 4 presents the most straightforward approach to deal with it, i.e., a general NLP model that describes the required orbital propagations and transfers. In turn, the Chapters contained in Part ii present more sophisticated techniques to compute the required maneuvers. Chapter 2 takes advantage of the properties of duality to efficiently solve specific instances of multi-impulse maneuvers, while guaranteeing the convergence and the global optimality of the solutions. Chapter 3 presents a methodology to compute J_2-perturbed low-thrust transfers between circular orbits that achieves an advantageous trade-off between the fidelity of the orbital dynamics, the optimality of the transfers and the computational efficiency.

Regarding the the interaction between the combinatorial decisions and the orbital dynamics, once again Chapter 4 presents the most straightforward approach to deal with it, i.e., a two-stage framework. Conversely, Chapter 5 takes advantage of the sequence information introduced by the no-subtour formulation to configure an integrated MILP model that seamlessly coordinates the maneuver optimization and object selection. However, the flexibility of the two-stage approach has proven very useful to configure the resolution of problems with a complicated structure, as shown in Chapters 6 and 7.

All in all, Chapter 4 is the cornestone of this thesis and constitutes the basis that has made it possible for the techniques explained in the rest of the chapters to flourish.

8.2 FUTURE WORK

Finally, the scope of the content presented in this thesis can be potentially enhanced, hence giving rise to further research opportunities.

Regarding Chapter 2, the use of the dual-based methodology to solve problems with a more general mathematical structure merits a deeper investigation, as it could lead to richer application cases.

Regarding Chapter 3, the possibility of solving optimal control problems by determining the initial values of physically meaningful variables, instead of the values of the costates, is definitely of great interest. Moreover, it is also worth to analyze the mathematical conditions under which the direct modification of the RAAN is more advantageous than increasing the nodal precession for J_2-perturbed transfers.

Regarding Chapter 4, the no-subtour formulation can potentially be applied for the modelling of numerous application problems in which the appearance of subtours is a concern or the problem constraints require sequence information to be formulated.

Regarding Chapter 5, it would be interesting to study the piecewise-linear approximation of functions with a larger number of variables, as well as its impact in the memory requirements of the resulting mathematical models.

Regarding Chapter 6, the use of a fuel station deserves additional exploration from a mission analysis perspective. A particularly compelling approach would be the use of a more permanent station, such that it could serve several debris removal missions within a large debris cluster.

Regarding Chapter 7, the proposed Constraint Programming framework can be generally applied to analyze other missions.

BIBLIOGRAPHY

[1] G. S. Aglietti, B. Taylor, S. Fellowes, T. Salmon, I. Retat, A. Hall, T. Chabot, A. Pisseloup, C. Cox, A. Mafficini, et al. "The active space debris removal mission RemoveDebris. Part 2: In orbit operations." In: *Acta Astronautica* 168 (2020), pp. 310–322.

[2] K. Apt. *Principles of constraint programming*. Cambridge university press, 2003.

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

[4] A. Balakrishnan, T. L. Magnanti, and P. Mirchandani. "A dual-based algorithm for multi-level network design." In: *Management science* 40.5 (1994), pp. 567–581. DOI: 10.1287/mnsc.40.5.567.

[5] J. Bang and J. Ahn. "Multitarget rendezvous for active debris removal using multiple spacecraft." In: *Journal of Spacecraft and Rockets* 56.4 (2019), pp. 1237–1247.

[6] B. W. Barbee, S. Alfano, E. Pinon, K. Gold, and D. Gaylor. "Design of Spacecraft Missions to Remove Multiple Orbital Debris Objects." In: *Guidance and Control 2012* 144 (2012), pp. 93–110.

[7] A. Barea, J.L. Gonzalo, C. Colombo, and H. Urrutxua. *A constraint programming framework for preliminary mission analysis: Applications for constellation-servicing active debris removal*. Submitted for publication.

[8] A. Barea, H. Urrutxua, and L. Cadarso. "Large-scale object selection and trajectory planning for multi-target space debris removal missions." In: *Acta Astronautica* 170.1 (2020), pp. 289–301. DOI: 10.1016/j.actaastro.2020.01.032.

[9] A. Barea, H. Urrutxua, and L. Cadarso. "A branch-and-bound method for the global optimization of active debris removal missions." In: *3rd IAA Conference on Space Situational Awareness (ICSSA)*. 2022, pp. 1–10.

[10] A. Barea, H. Urrutxua, and L. Cadarso. "Dual-Based Method for Global Optimization of Impulsive Orbital Maneuvers." In: *The Journal of the Astronautical Sciences* (2022), pp. 1–26. DOI: 10.1007/s40295-022-00357-5.

[11] A. Barea, H. Urrutxua, and L. Cadarso. "Preliminary Analysis of a Fuel Station Strategy for Active Debris Removal Missions in Low Earth Orbit." In: *2022 AAS/AIAA Astrodynamics Specialist Conference*. 2022, pp. 1–12.

[12] A. Barea, H. Urrutxua, and L. Cadarso. "Relative Inclination Strategy for J2-Perturbed Low-Thrust Transfers Between Circular Orbits." In: *Journal of Guidance, Control, and Dynamics* (2022). DOI: 10.2514/1.G006755.

[13] J. Barreiro, M. Boyce, M. Do, J. Frank, M. Iatauro, T. Kichkaylo, P. Morris, J. Ong, E. Remolina, T. Smith, et al. "EUROPA: A platform for AI planning, scheduling, constraint programming, and optimization." In: *4th International Competition on Knowledge Engineering for Planning and Scheduling (ICKEPS)* (2012).

[14] R. H. Battin. *An Introduction to the Mathematics and Methods of Astrodynamics, revised edition*. American Institute of Aeronautics and Astronautics, 1999.

[15] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty. *Nonlinear programming: theory and algorithms*. Hoboken: John Wiley & Sons, 2013.

[16] E.M.L. Beale and John J.H. Forrest. "Global optimization using special ordered sets." In: *Mathematical Programming* 10.1 (1976), pp. 52–69.

[17] J. F. Benders. "Partitioning procedures for solving mixed-variables programming problems." In: *Numerische Mathematik* 4.1 (1962), pp. 238–252. DOI: 10.1007/BF01386316.

[18] N. Berend and X. Olive. "Bi-objective optimization of a multiple-target active debris removal mission." In: *Acta Astronautica* 122 (2016), pp. 324–335. DOI: 10.1016/j.actaastro.2016.02.005.

[19] R. Biesbroek, S. Aziz, A. Wolahan, S. Cipolla, M. Richard-Noca, and L. Piguet. "The clearspace-1 mission: Esa and clearspace team up to remove debris." In: *Proc. 8th Eur. Conf. Sp. Debris*. 2021, pp. 1–3.

[20] C. Blackerby, A. Okamoto, K. Fujimoto, N. Okada, J. L. Forshaw, and J. Auburn. "ELSA-d: An In-Orbit End-of-Life Demonstration Mission." In: *Proceedings of the 69th International Astronautical Congress, Bremen, Germany: International Astronautical Federation*. 2018.

[21] C. Blackerby, A. Okamoto, S. Iizuka, Y. Kobayashi, K. Fujimoto, Y. Seto, S. Fujita, T. Iwai, N. Okada, J. Forshaw, et al. "The ELSA-d end-of-life debris removal mission: preparing for launch." In: *Proceedings of the International Astronautical Congress, IAC*. Vol. 8. 2019.

[22] G. Borelli, G. Gaias, C. Colombo, and L. Vallini. "Rendezvous and Proximity Operations Design of an Active Debris Removal Service to a Large Constellation Fleet." In: *72nd International Astronautical Congress (IAC 2021)*. 2021, pp. 1–13.

[23] G. Borelli, M. Trisolini, M. Massari, and C. Colombo. "A Comprehensive Ranking Framework for Active Debris Removal Missions Candidates." In: *8th European Conference on Space Debris, ESA/ESOC*. ESA. 2021, pp. 1–10.

[24] R. P. Brent. "An algorithm with guaranteed convergence for finding a zero of a function." In: *The Computer Journal* 14.4 (1971), pp. 422–425. ISSN: 0010-4620. DOI: 10.1093/comjnl/14.4.422.

[25] H. Brettle, H. Lewis, T. Harris, and M. Lindsay. "Assessing Debris Removal Services for Large Constellations." In: *8th European Conference on Space Debris, ESA/ESOC*. ESA. 2021, pp. 1–8.

[26] F. J. de Bruijn, S. Theil, D. Choukroun, and E. Gill. "Collocation of geostationary satellites using convex optimization." In: *Journal of Guidance, Control, and Dynamics* 39.6 (2016), pp. 1303–1313. DOI: 10.2514/1.G001650.

[27] L. Casalino and G. Colasurdo. "Improved Edelbaum's approach to optimize low earth/geostationary orbits low-thrust transfers." In: *Journal of guidance, control, and dynamics* 30.5 (2007), pp. 1504–1511. DOI: 10.2514/1.28694.

[28] L. Casalino and D. Pastrone. "Active debris removal missions with multiple targets." In: *AIAA/AAS Astrodynamics Specialist Conference*. 2014.

[29] CelesTrack. *Two Line Element set for Starlink constellation*. https://celestrak.com/NORAD/elements/starlink.txt. Accessed: 21 January 2022. 2022.

[30] 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. DOI: 10.1007/s10957-012-0130-6.

[31] M. Cerf. "Multiple Space Debris Collecting Mission: Optimal Mission Planning." In: *Journal of Optimization Theory and Applications* 167.1 (2015), pp. 195–218. DOI: 10.1007/s10957-015-0705-0.

[32] M. Cerf. "Low-thrust transfer between circular orbits using natural precession." In: *Journal of Guidance, Control, and Dynamics* 39.10 (2016), pp. 2232–2239. DOI: 10.2514/1.G001331.

[33] S. Chien, M. Johnston, J. Frank, M. Giuliano, A. Kavelaars, C. Lenzen, and N. Policella. "A generalized timeline representation, services, and interface for automating space mission operations." In: *SpaceOps 2012*. 2012, pp. 1–17.

[34] V. A. Chobotov. *Orbital mechanics*. Reston: American Institute of Aeronautics and Astronautics, 2002.

[35] C. Colombo, S. Huang, G. Borelli, F. Cavenago, M. Nugnes, J.L. Gonzalo, G. Gaias, M. Massari, L. Vallini, M. Petit, et al. "Mission analysis and design for an active debris removal service for large constellations." In: *8th European Conference on Space Debris, ESA/ESOC*. ESA. 2021, pp. 1–11.

[36] Federal Communications Commission. *Mitigating Orbital Debris by Shortening Time for Satellite Disposal*. IB Docket Nos. 22-271 and 18-313. 2022.

[37] Inter-Agency Space Debris Coordination Committee. *IADC Space Debris Mitigation Guidelines*. IADC-02-01 Rev. 3. 2021.

[38] K. Daneshjou, A. A. Mohammadi-Dehabadi, and M. Bakhtiari. "Mission planning for on-orbit servicing through multiple servicing satellites: A new approach." In: *Advances in Space Research* 60.6 (2017), pp. 1148–1162. DOI: 10.1016/j.asr.2017.05.037.

[39] G. Dantzig, R. Fulkerson, and S. Johnson. "Solution of a large-scale traveling-salesman problem." In: *Journal of the operations research society of America* 2.4 (1954), pp. 393–410.

[40] M. Di Carlo, J. M. Romero Martin, and M. Vasile. "Automatic trajectory planning for low-thrust active removal mission in low-earth orbit." In: *Advances in Space Research* 59.5 (2017), pp. 1234–1258. ISSN: 0273-1177. DOI: 10.1016/j.asr.2016.11.033.

[41] T. N. Edelbaum. "Propulsion requirements for controllable satellites." In: *Ars Journal* 31.8 (1961), pp. 1079–1089. DOI: 10.2514/8.5723.

[42] D. Erlenkotter. "A dual-based procedure for uncapacitated facility location." In: *Operations Research* 26.6 (1978), pp. 992–1009. DOI: 10.1287/opre.26.6.992.

[43] L. Federici, A. Zavoli, and G. Colasurdo. "A Time-Dependent TSP Formulation for the Design of an Active Debris Removal Mission using Simulated Annealing." In: *AIAA/AAS Astrodynamics Specialist Conference*. 2019.

[44] L. Federici, A. Zavoli, and G. Colasurdo. "Impulsive Multi-Rendezvous Trajectory Design and Optimization." In: *European Conference for Aeronautics and Space Sciences*. 2019.

[45] L. Federici, A. Zavoli, and G. Colasurdo. "On the use of A* search for active debris removal mission planning." In: *Journal of Space Safety Engineering* 8.3 (2021), pp. 245–255.

[46] J. L. Forshaw, G. S. Aglietti, S. Fellowes, T. Salmon, I. Retat, A. Hall, T. Chabot, A. Pisseloup, D. Tye, C. Bernal, et al. "The active space debris removal mission RemoveDebris. Part 1: From concept to launch." In: *Acta Astronautica* 168 (2020), pp. 293–309.

[47] J. Forshaw, R. Vos van Steenwijk, S. Wokes, S. Ainley, A. Bradford, J. Auburn, C. Blackerby, and N. Okada. "Preliminary Design of an End-of-life ADR Mission for Large Constellations." In: *70th International Astronautical Congress (IAC), Washington DC, US*. 2019, pp. 21–25.

[48] K. Fox, B. Gavish, and S. Graves. "An N-Constraint Formulation of the (Time-Dependent) Traveling Salesman Problem." In: *Operations Research* 28 (1980), pp. 1018–1021. DOI: 10.1287/opre.28.4.1018.

[49] J. Frank, A. Jonsson, R. Morris, D. E. Smith, and P. Norvig. "Planning and scheduling for fleets of earth observing satellites." In: *International Symposium on Artificial Intelligence, Robotics, Automation and Space*. 2001.

[50] G. Gatto and L. Casalino. "Fast evaluation and optimization of low-thrust transfers to multiple targets." In: *Journal of Guidance, Control, and Dynamics* 38.8 (2015), pp. 1525–1530. DOI: 10.2514/1.G001116.

[51] A. M. Geoffrion. "Duality in nonlinear programming: a simplified applications-oriented development." In: *SIAM review* 13.1 (1971), pp. 1–37. DOI: 10.1137/1013001.

[52] A. M. Geoffrion. "Generalized Benders decomposition." In: *Journal of Optimization Theory and Applications* 10.4 (1972), pp. 237–260. DOI: 10.1007/BF00934810.

[53] J. L. Gonzalo, C. Colombo, and P. Di Lizia. "Analytical framework for space debris collision avoidance maneuver design." In: *Journal of Guidance, Control, and Dynamics* 44.3 (2021), pp. 469–487.

[54] M. Guignard and M. B. Rosenwein. "An improved dual based algorithm for the generalized assignment problem." In: *Operations Research* 37.4 (1989), pp. 658–663. DOI: 10.1287/opre.37.4.658.

[55] P.-E. Hladik, H. Cambazard, A.-M. Deplanche, and N. Jussien. "Solving a real-time allocation problem with constraint programming." In: *Journal of Systems and Software* 81.1 (2008), pp. 132–149.

[56] T. D. Hoang, L. B. Le, and T. Le-Ngoc. "Energy-efficient resource allocation for D2D communications in cellular networks." In: *IEEE Transactions on vehicular technology* 65.9 (2015), pp. 6972–6986. DOI: 10.1109/TVT.2015.2482388.

[57] S. Huang, C. Colombo, J.L. Gonzalo, A. Masserini, M. Nugnes, L. Vallini, and M. Petit. "Preliminary Mission Analysis of Active Debris Removal Service for Large Constellations." In: *71st International Astronautical Congress (IAC 2020)*. 2020, pp. 1–6.

[58] D. Izzo, I. Getzner, D. Hennes, and L. F. Simoes. "Evolving Solutions to TSP Variants for Active Space Debris Removal." In: *Gecco'15: Proceedings of the 2015 Genetic and Evolutionary Computation Conference* (2015), pp. 1207–1214. DOI: 10.1145/2739480.2754727.

[59] X. Jiang and R. Xu. "A constraint-programmed planner for deep space exploration problems with table constraints." In: *IEEE Access* 5 (2017), pp. 17258–17270.

[60] N. Johnson. "History and consequences of on-orbit break-ups." In: *Advances in Space Research* 5.2 (1985), pp. 11–19.

[61] N. Johnson. "Origin of the Inter-Agency Space Debris Coordination Committee." In: *ARES Biennial Report 2012 Final* (2014).

[62] J. A. Kechichian. "Reformulation of Edelbaum's low-thrust transfer problem using optimal control theory." In: *Journal of Guidance, Control, and Dynamics* 20.5 (1997), pp. 988–994. DOI: 10.2514/2.4145.

[63] T.S. Kelso. "Analysis of the 2007 Chinese ASAT Test and the Impact of its Debris on the Space Environment." In: *8th Advanced Maui Optical and Space Surveillance Technologies Conference, Maui, HI*. Vol. 7. 2007.

[64] T.S. Kelso. "Analysis of the Iridium 33-Cosmos 2251 collision." In: *Advances in the Astronautical Sciences* 135.2 (2009), pp. 1099–1112.

[65] D. J. Kessler and B. G. Courpalais. "Collision Frequency of Artificial Satellites - Creation of a Debris Belt." In: *Journal of Geophysical Research-Space Physics* 83.A6 (1978), pp. 2637–2646. DOI: 10.1029/JA083iA06p02637.

[66] C. A. Kluever. "Using edelbaum's method to compute Low-Thrust transfers with earth-shadow eclipses." In: *Journal of Guidance, Control, and Dynamics* 34.1 (2011), pp. 300–303. DOI: 10.2514/1.51024.

[67] R. Knight, C. Chouinard, G. Jones, and D. Tran. "Leveraging multiple artificial intelligence techniques to improve the responsiveness in operations planning: Aspen for orbital express." In: *AI Magazine* 35.4 (2014), pp. 26–36.

[68] Kuiper Systems LLC. *Request for Experimental Authorization*. https://apps.fcc.gov/els/GetAtt.html?id=285359. Accessed: 8 November 2022. 2022.

[69] A. Langevin, F. Soumis, and J. Desrosiers. "Classification of travelling salesman problem formulations." In: *Operations Research Letters* 9.2 (1990), pp. 127–132.

[70] B. Larbi, M.K. Grzesik, B. Radtke, C.J. Trentlage, and E. Stoll. "Active debris removal for mega constellations: Cubesat possible." In: *Proceedings of the 9th International Workshop on Satellite Constellations and Formation Flying IWSCFF2017, Boulder, Colorado*. 2017.

[71] H. G. Lewis, A. E. White, R. Crowther, and H. Stokes. "Synergy of debris mitigation and removal." In: *Acta Astronautica* 81.1 (2012), pp. 62–68. DOI: 10.1016/j.actaastro.2012.06.012.

[72] R. M. Lewis and V. Torczon. "Pattern search methods for linearly constrained minimization." In: *SIAM Journal on Optimization* 10.3 (2000), pp. 917–941.

[73] H. Li, S. Chen, and H. Baoyin. "J2-Perturbed Multitarget Rendezvous Optimization with Low Thrust." In: *Journal of Guidance Control and Dynamics* 41.3 (2018), pp. 802–808. DOI: 10.2514/1.G002889.

[74] H. Li, S. Chen, and H. Baoyin. "J2-perturbed multitarget rendezvous optimization with low thrust." In: *Journal of Guidance, Control, and Dynamics* 41.3 (2018), pp. 802–808. DOI: 10.2514/1.G002889.

[75] J.-C. Liou and N. L. Johnson. "A sensitivity study of the effectiveness of active debris removal in LEO." In: *Acta Astronautica* 64.2-3 (2009), pp. 236–243. DOI: 10.1016/j.actaastro.2008.07.009.

[76] Y. Liu, J. Yang, Y. Wang, Q. Pan, and J. Yuan. "Multi-objective optimal preliminary planning of multi-debris active removal mission in LEO." In: *Science China-Information Sciences* 60.7 (2017), p. 072202. DOI: 10.1007/s11432-016-0566-7.

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

[78] S. Martello and P. Toth. *Knapsack problems: algorithms and computer implementations*. Hoboken: John Wiley & Sons, Inc., 1990.

[79] L. S. Maxeiner and S. Engell. "Comparison of dual based optimization methods for distributed trajectory optimization of coupled semi-batch processes." In: *Optimization and Engineering* 21.3 (2020), pp. 761–802. DOI: 10.1007/s11081-020-09499-7.

[80] C. E. Miller, A. W. Tucker, and R. A. Zemlin. "Integer programming formulation of traveling salesman problems." In: *Journal of the ACM (JACM)* 7.4 (1960), pp. 326–329.

[81] C. E. Miller, A. W. Tucker, and R. A. Zemlin. "Integer programming formulation of traveling salesman problems." In: *Journal of the ACM (JACM)* 7.4 (1960), pp. 326–329.

[82] R. Misener and C.A. Floudas. "Piecewise-linear approximations of multidimensional functions." In: *Journal of optimization theory and applications* 145.1 (2010), pp. 120–147.

[83] J. Murakami and S. Hokamoto. "Approach for optimal multi-rendezvous trajectory design for active debris removal." In: *61st International Astronautical Congress 2010, IAC 2010*. Vol. 7. 2010, pp. 6013–6018.

[84] Network Access Associates Ltd. *OneWeb webpage*. https://oneweb.net/. Accessed: 8 November 2022. 2022.

[85] J. T. Olympio and N. Frouvelle. "Space debris selection and optimal guidance for removal in the SSO with low-thrust propulsion." In: *Acta Astronautica* 99 (2014), pp. 263–275. DOI: 10.1016/j.actaastro.2014.03.005.

[86] C. Pardini and L. Anselmo. "Characterization of abandoned rocket body families for active removal." In: *Acta Astronautica* 126 (2016). Space Flight Safety, pp. 243 –257. ISSN: 0094-5765. DOI: 10.1016/j.actaastro.2016.04.035.

[87] J. C. Pemberton and F. Galiber. "A constraint-based approach to satellite scheduling." In: *DIMACS Series in Discrete Mathematics and Theoretical Computer Science* 57 (2001), pp. 101–114.

[88] J.C. Pemberton. "Towards scheduling over-constrained remote sensing satellites." In: *Proceedings of the 2d International Workshop on Planning and Scheduling for Space*. 2000.

[89] G. Pesant, M. Gendreau, J.-Y. Potvin, and J.-M. Rousseau. "On the flexibility of constraint programming models: From single to multiple time windows for the traveling salesman problem." In: *European Journal of Operational Research* 117.2 (1999), pp. 253–263.

[90] J. Rodriguez. "A constraint programming model for real-time train scheduling at junctions." In: *Transportation Research Part B: Methodological* 41.2 (2007), pp. 231–245.

[91] A. Rossi, G. B. Valsecchi, and E. M. Alessi. "The Criticality of Spacecraft Index." In: *Advances in Space Research* 56.3 (2015). Advances in Asteroid and Space Debris Science and Technology - Part 1, pp. 449 –460. ISSN: 0273-1177. DOI: 10.1016/j.asr.2015.02.027.

[92] R.R.K. Sharma and K.D. Sharma. "A new dual based procedure for the transportation problem." In: *European Journal of Operational Research* 122.3 (2000), pp. 611–624. DOI: 10.1016/S0377-2217(99)00081-8.

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

[94] H.-X. Shen. "Explicit Approximation for J2-Perturbed Low-Thrust Transfers Between Circular Orbits." In: *Journal of Guidance, Control, and Dynamics* 44.8 (2021), 1525–1531. DOI: 10.2514/1.G005415.

[95] H.-X. Shen, T.-J. Zhang, L. Casalino, and D. Pastrone. "Optimization of Active Debris Removal Missions with Multiple Targets." In: *Journal of Spacecraft and Rockets* 55.1 (2018), pp. 181–189. DOI: 10.2514/1.A33883.

[96] H. Shen and P. Tsiotras. "Peer-to-peer refueling for circular satellite constellations." In: *Journal of Guidance, Control, and Dynamics* 28.6 (2005), pp. 1220–1230. DOI: 10.2514/1.9570.

[97] G. Simonin, C. Artigues, E. Hebrard, and P. Lopez. "Scheduling scientific experiments on the Rosetta/Philae mission." In: *International Conference on Principles and Practice of Constraint Programming*. Springer. 2012, pp. 23–37.

[98] Starlink Services LLC. *Starlink webpage*. https://www.starlink.com/. Accessed: 8 November 2022. 2022.

[99] J. Stuart, K. Howell, and R. Wilson. "Application of multi-agent coordination methods to the design of space debris mitigation tours." In: *Advances in Space Research* 57.8 (2016), pp. 1680–1697. DOI: 10.1016/j.asr.2015.05.002.

[100] D. A. Vallado. *Fundamentals of astrodynamics and applications*. Vol. 12. El Segundo: Springer Science & Business Media, 2001.

[101] M. Wallace. "Practical applications of constraint programming." In: *Constraints* 1.1 (1996), pp. 139–168.

[102] Y. Wang, G. Zhou, L. Caccetta, and W. Liu. "An alternative Lagrange-dual based algorithm for sparse signal reconstruction." In: *IEEE transactions on signal processing* 59.4 (2010), pp. 1895–1901. DOI: 10.1109/TSP.2010.2103066.

[103] B. Weeden. "Overview of the legal and policy challenges of orbital debris removal." In: *Space Policy* 27.1 (2011), pp. 38–43.

[104] C. Wen, C. Zhang, Y. Cheng, and D. Qiao. "Low-Thrust Transfer Between Circular Orbits Using Natural Precession and Yaw Switch Steering." In: *Journal of Guidance, Control, and Dynamics* 44.7 (2021), 1371–1378. DOI: 10.2514/1.G005790.

[105] J. Yang, Y. Hen Hu, Y. Liu, and Q. Pan. "A maximal-reward preliminary planning for multi-debris active removal mission in LEO with a greedy heuristic method." In: *Acta Astronautica* 149 (2018), pp. 123–142. DOI: 10.1016/j.actaastro.2018.05.040.

[106] J. Yu, X.-Q. Chen, and L.-H. Chen. "Biobjective planning of GEO debris removal mission with multiple servicing spacecrafts." In: *Acta Astronautica* 105.1 (2014), pp. 311–320. DOI: 10.1016/j.actaastro.2014.10.003.

[107] J. Yu, X.-Q. Chen, and L.-H. Chen. "Optimal planning of LEO active debris removal based on hybrid optimal control theory." In: *Advances in Space Research* 55.11 (2015), pp. 2628–2640. DOI: 10.1016/j.asr.2015.02.026.

[108] J. Yu, X.-Q. Chen, L.-H. Chen, and D. Hao. "Optimal scheduling of GEO debris removing based on hybrid optimal control theory." In: *Acta Astronautica* 93 (2014), pp. 400–409. DOI: 10.1016/j.actaastro.2013.07.015.

[109] J. Zhang, G. T. Parks, Y.-Z. Luo, and G.-J. Tang. "Multispacecraft refueling optimization considering the J2 perturbation and window constraints." In: *Journal of Guidance, Control, and Dynamics* 37.1 (2014), pp. 111–122. DOI: 10.2514/1.61812.

[110] T.-J. Zhang, Y.-K. Yang, B.-H. Wang, Z. Li, H.-X. Shen, and H.-N. Li. "Optimal scheduling for location geosynchronous satellites refueling problem." In: *Acta Astronautica* 163 (2019), pp. 264–271.

[111] Z. Zhao, J. Zhang, H.-Y. Li, and J.-Y. Zhou. "LEO cooperative multi-spacecraft refueling mission optimization considering J2 perturbation and target's surplus propellant constraint." In: *Advances in Space Research* 59.1 (2017), pp. 252–262. DOI: 10.1016/j.asr.2016.10.005.

[112] Y. Zhou, Y. Yan, X. Huang, and L. Kong. "Mission planning optimization for multiple geosynchronous satellites refueling." In: *Advances in Space Research* 56.11 (2015), pp. 2612–2625.

[113] Y. Zhou, Y. Yan, X. Huang, and L. Kong. "Optimal scheduling of multiple geosynchronous satellites refueling based on a hybrid particle swarm optimizer." In: *Aerospace Science and Technology* 47 (2015), pp. 125–134.

[114] X. Zhu, J. Chen, C. Zhang, and B. Qiao. "Optimal fuel station arrangement for multiple GEO spacecraft refueling mission." In: *Advances in Space Research* 66.8 (2020), pp. 1924–1936.

[115] F. Zuiani and M. Vasile. "Preliminary Design of Debris Removal Missions by Means of Simplified Models for Low-Thrust, Many-Revolution Transfers." In: *International Journal of Aerospace Engineering* 2012 (2012). DOI: 10.1155/2012/836250.