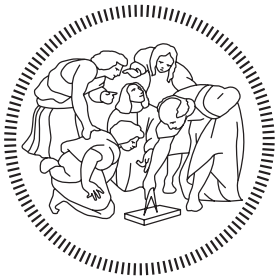
School of Industrial and Information Engineering
Department of Aerospace Science and Technology

Master of Science in
SPACE ENGINEERING



POLITECNICO
MILANO 1863

Routing optimization for impulsive and low-thrust multi-injection problem solving for constellations and multi-satellites deployment

SUPERVISOR:
Professor Michèle Lavagna

CO-SUPERVISOR:
Mr. Jacopo Prinetto

CANDIDATE:
Vincenzo Maria Salvato
920779

A.Y. 2019-2020

Don't panic.

Abstract

THE continuously growing number of small satellites which needs to be launched around Earth nowadays [1] [2] asks for new releasing strategies. This work suggests a new approach to solve the routing problem [3] [4] and fast identify feasible profiles to simultaneously and efficiently inject into orbit multiple space assets by means of a deployer equipped with low-thrust control authority. The possibility to perform the transfers with impulsive maneuvers was also investigated. A multi-satellites single launch injection scenario asks solving two main challenges from the dynamics control management perspective: fast converge on a feasible release history to maximise the launcher utilisation and revenue; grant flexibility and robustness in managing on orbit operations quick reshaping along the whole time window devoted to the complete set of satellites injection. Therefore, the proposed algorithm, while finding the near-optimal releasing order to deploy N heterogeneous satellites correctly phased on their operational orbits, aims of being computationally light and fast. The algorithm is proposed to be as flexible as possible in terms of scenarios, being compatible with both single-launch homogeneous space assets constellation and heterogeneous multi-satellites deployment, differing in final orbit insertion and physical properties. A multi-objective heuristic optimization is here preferred to find the optimal releasing order and transfer strategies, aiming to minimize the fuel consumption and the time to operations. Furthermore, the algorithm easily manages the introduction of engineering constraints, such as maximum thrust or power available, and of operative constraints, such as scheduling constraints. The most demanding transfers in term of propellant and time, which are those entailing large plane changes (i.e. RAAN and inclination), are performed through the exploitation of the asymmetrical Earth gravitational field [5]. This approach leads to consistent savings in propellant, at the cost of increase in overall time of deployment, granting the great flexibility of the algorithm in terms of applications, allowing performing several transfers connecting orbits with very different parameters. Performances of the algorithm are tested when stressed in terms of both search space and constraints set size.

Keywords: low-thrust; impulsive; routing; optimization; constellation; multi-mission; multi-injection; multi-deployment.

Estratto

La continua crescita del numero di satelliti di piccole dimensioni che necessitano di essere lanciati oggigiorno [1] [2] richiede nuove strategie di rilascio. Questo lavoro propone un nuovo approccio per risolvere il problema di routing [3] [4] ed identificare velocemente dei possibili profili per rilasciare simultaneamente ed efficacemente in orbita più satelliti per mezzo di un dispenser equipaggiato con propulsori a bassa spinta. La possibilità di effettuare i trasferimenti attraverso manovre impulsive è anche presa in considerazione. Il planning di una missione di multi-rilascio con un singolo lancio richiede la soluzione di due principali problemi: convergere velocemente su un ordine di rilascio fattibile che massimizzi l'utilizzo del lanciatore ed il guadagno; garantire flessibilità e robustezza nel gestire operazioni on orbit durante tutta la finestra temporale della missione. L'algoritmo proposto punta a trovare una traiettoria sub-ottimale per il rilascio di N satelliti e correttamente distribuiti sulle rispettive orbite operazionali e ad essere computazionalmente leggero e veloce. L'algoritmo si propone inoltre di essere il più flessibile possibile in termini di scenari di applicazione, essendo compatibile con rilascio multiplo di satelliti sia omogenei, ovvero con caratteristiche simili sia in orbita di rilascio che in proprietà fisiche, che eterogenei. Un metodo di ottimizzazione euristico multi-obiettivo è stato preferito per trovare l'ordine di rilascio sub-ottimale e le singole strategie di trasferimento, con l'obiettivo di minimizzare il consumo di propellente e la durata della missione. Inoltre, l'algoritmo garantisce la possibilità di introdurre vincoli ingegneristici, quali la massima spinta o massima potenza disponibile, o operativi, come sull'ordine di rilascio. I trasferimenti più impegnativi in termini di consumo di propellente e durata, ovvero quelli che comprendono grandi cambi di piano (i.e. RAAN e inclinazione), sono eseguiti sfruttando l'asimmetria del campo gravitazionale terrestre [5]. Questo approccio porta a significativi risparmi nel consumo di propellente al costo di un aumento della durata della missione, garantendo così la grande flessibilità dell'algoritmo in termini di applicazione. Le performance dell'algoritmo sono testate quando sottoposto a condizioni limite sia in termini di dimensioni dello spazio di ricerca che di vincoli.

Parole chiave: bassa spinta; impulsivo; routing; ottimizzazione; costellazione; multi-satellite; multi-rilascio.

Contents

List of Figures

List of Tables

Acronyms

ADR	Active Debris Removal
EA	Evolutionary Algorithms
GA	Genetic Algorithm
GEO	Geosynchronous Equatorial Orbit
GTOC	Global Trajectories Optimization Competition
HOCP	Hybrid Optimal Control Problem
JD	Julian Date
KP	Keplerian Parameters
LEO	Low Earth Orbit
LTAN	Local Time of the Ascending Node
MEE	Modified Equinoctial Elements
MGA	Multiple Gravity Assists
MINLP	Mixed-Integer Nonlinear Programming
MJD	Modified Julian Date
MOPSO	Multi-Objective Particle Swarm Optimization
PSO	Particle Swarm Optimization
RAAN	Right Ascension of the Ascending Node
SI	Swarm Intelligence
SSO	Sun-Synchronous Orbit
TPBVP	Two-Points Boundary Values Problem
TOF	Time Of Flight

TPA Two Phase Algorithm

TSP Travelling Salesman Problem

VRP Vehicle Routing Problem

1. Introduction

1.1 Motivation of the research

INTEREST in space and space applications is growing. With the advent of CubeSats and SmallSats, which range from 0.01 to 180 kg [6], the need for new techniques to launch in space these classes of satellites is arising. In the incoming years, a substantial increment in the number of small satellites to be launched has been forecasted [2]. Studies suggest that the number of satellites launched in the decade 2019-2028 will have a x4 growth rate compared to the previous decade and that satellites with a launch mass < 500 kg will account for 87% of such number. In addition to this, it was found that "despite a growing number of operational dedicated launch vehicles, the majority of nano/microsatellites in 2019 chose to leverage rideshare alternatives" [1]. The drawback of piggyback launches is in the fact that usually the small satellites are released on the target orbit of the main payload or in its proximity. Consequently, these satellites would need their propulsive system to be capable to allocate themselves on the correct orbit and with the desired phasing. Also, they would need to wait for a launch whose main payload has a target orbit as similar as possible to their final one.

Being such satellites the largest market share, new ways to facilitate their access to space are being investigated. A possible solution to overcome the drawbacks of piggyback payload launches is to develop the last stages of launchers or dedicated vehicles able to carry the small satellites directly on their operational orbit. Some of such vehicles already exist, such as the Small Launch Orbital Maneuvering Vehicle (SL-OMV), as a propulsive tug for secondary payload deployment equipped with green monopropellant thrusters, developed by Moog Inc. [7]. Spaceflight Inc. is working at the Sherpa-NG (Next Generation) program, a new program to release smallsats to custom orbital destinations. The company has already developed and successfully deployed 14 satellites (January 2021) with the Sherpa-FX and is planning to launch also the Sherpa-LTE, a next-generation transfer vehicle capable of multiple deployments based on electric propulsion [8]. In addition to them, the Italian company D-Orbit has developed the ION (InOrbit Now) satellite carrier [9], a dispenser that transports spacecraft to the desired

operational orbits and deploys them at the correct phasing.

1.2 Contribution of the work

While most of the existing or under study multi-deploying vehicles rely on chemical propulsion, one interesting way to approach the problem may be considering a low-thrust vehicle to release several satellites in their respective operational orbits. Previous works dealing with routing problems mainly focused on Active Debris Removal (ADR) [10] [11] [12] [13], on-orbit servicing [14] [15] or Multiple Gravity Assists (MGA) [16]. The problem of small-satellites multi-platform insertion is different from ADR missions since the former is characterized by a negative variation in dry mass after each release. MGA missions are different for the application scenario since do not take place around Earth but in outer space.

Only a few of the previous studies about routing problems focused on low-thrust. The introduction of low-thrust instead of impulsive maneuvers makes this work different from the vast majority of previous studies about multi-deployment. This new approach leads to a substantial increase in the complexity of the problem since keplerian dynamics is now affected by the thrust for the whole duration of the orbital transfers. Also, natural disturbances must be taken into account to achieve good results since they are in some cases about the same order of magnitude of the low-thrust propulsion if not higher. In particular, this work will consider an upper stage of a launcher, provided with electric primary propulsion and in charge of the multi-delivery

This research wants to develop an algorithm to optimize the combination of the low-thrust trajectories between the different release orbits around Earth. In particular, this work focuses on developing an algorithm that is able to give an estimate of the duration and of the propellant consumption of a multi-deployment mission in a fast and efficient way. Keeping the computational cost of the algorithm low is of paramount importance to allow the exploration of numerous routing possibilities in a small amount of time. Given its significant value, the computational effort of the algorithm will be the main driver in the algorithm development and hence affecting most of the choices behind it.

The problem of deploying several satellites in different positions around Earth can be seen as a variant of the Travelling Salesman Problem (TSP), a well-known problem of operations research, which is a particular case of the more general Vehicle Routing Problem (VRP). The main difference with the usual formulation of the problem is that, in its application to space dynamics, the destinations are not fixed in time and space anymore. The slots in which the satellites have to be released are indeed dependent on the epoch. A heuristic optimization method will be selected to find the sub-optimal

solution of the VRP [11] [12] [14]. Indeed, due to the very large search space characterizing the problem of releasing satellites on a high number of different orbits, there is the need to choose an optimization method that allows finding optimal or near-optimal solutions with low computational effort. This is indeed necessary to compare different options and efficiently plan the multi-deployment mission.

1.3 State of the art

This section presents the current level of development of the three main pillars of the thesis:

1. First, the state of the art of the studies about the VRP in the space environment is summarized.
2. Afterwards, an overview of the current development of optimization methods and in particular heuristic ones is reported.
3. Lastly, the methods in which the low-thrust trajectories can be computed, with particular focus on the shape-based methods, are presented.

1.3.1 Vehicle routing problem

Finding the optimal hopping path between different orbits means solving a variant of the TSP, which is a particular formulation of the VRP. The latter belongs to the class of the NP-hard problems, meaning that the computational time required to solve them dramatically increases with the size of the problem.

The application of the VRP to space is not an absolute novelty. One of the first applications can be found in the work by Alfriend et al. [14], where the optimal rendezvous with geosynchronous satellites is dealt with. The objective of the optimization is minimizing the velocity change Δv induced by out-of-plane maneuvers, considered the most significant part of the consumption. Most of the VRP applications to space have been about ADR, such as the work by Izzo et al. [12], where the VRP is approached through different methods, such as the inver-over algorithm or the tree search. First, the problem is treated as a static one, meaning with orbits fixed in space and time. Afterward, the dynamic variant is approached, where the orbits are not considered fixed anymore but perturbed by natural disturbances. Another interesting work is the one by Bèrend et al. [11], where instead branch and bound is used to define the visiting order of the different debris of the TSP. The previous works mainly focused on chemical-propelled spacecraft. Only a few VRP studies have focused on low-thrust, such as the one by Yam et al. [17], which deals with a low-thrust MGA problem.

Many problems in orbital mechanics, like this variant of the TSP, can be formulated as Hybrid Optimal Control Problem (HOCP). These problems include two different kinds of variables: continuous-valued variables, such as the ones describing the state of the spacecraft in time, and binary variables, such as the ones defining the visiting order of the hopping trajectory. In particular, this problem belongs to the Mixed-Integer Nonlinear Programming (MINLP), the area of optimization which deals with non-linear problems with continuous and integer variables. Alternatively, to solve the problem a two-layer optimization scheme can be adopted, as suggested by Conway and Wall [18]. There, an outer-loop genetic algorithm defines the visiting order while the inner-loop one locates near-optimal solutions for the trajectory, computes the cost and returns it to the outer-loop genetic algorithm. In this way, it can be avoided to deal simultaneously with both discrete and continuous-valued variables. These different options were investigated in the work by Zhang et al. [4], who showed that the two-level optimization presented the worst performance. In light of such results, this work focuses on finding a solution to the MINLP.

1.3.2 Heuristic methods

A mission scenario with numerous satellites to release would lead to a problem with a really large solution space. Being the VRP an NP-hard problem, this would lead to the need for a very high computational time to find the optimal solution. In addition to this, the research space in this kind of problem is not a simply connected domain. For these reasons, it is necessary to take into account optimization methods that trade optimality for speed. Heuristic methods [19] do not grant to find the optimal solutions to the problem but the low computational effort required to achieve this near-optimal solution makes them very valuable. Most complex problems like this variant of the VRP require the evaluation of an immense number of possibilities to determine an exact solution; the time required to find it may sometimes require even more than a lifetime. In addition to this, some problems such as the one dealt with in this thesis may require to be solved much more than one time, depending on their use and goal. For this reason, it is useful to have an algorithm whose solution can be found in a relatively short amount of time. Heuristics play an effective role in such problems by finding a way to reach solutions with reasonable computational effort.

In particular problem-independent algorithms are often referred to as "metaheuristic" algorithms. There is a number of metaheuristic algorithms that could be adopted to solve the problem and it is important to select a performing one. Each algorithm is characterized by a particular set of tuning parameters that could be optimally set to maximize its efficiency. In the last years, several metaheuristic methods have been developed and they can be classified in many ways. One of the most significant classification [20] is the

following:

- **Trajectory methods**: these algorithms work on one single solution at a time, describing a trajectory in the search space during the search process. They encompass local search-based metaheuristics.
- **Population-based methods**: these algorithms, on the contrary, deal in every iteration with a set of solutions, therefore providing a way to efficiently explore the search space.

While the former class of metaheuristics may enable better exploration of a promising area, the latter ones are more suitable to find promising areas in the search space [20]. Due to the vast solution space of the problem treated in this work, population-based methods will be preferred to the trajectory ones for this application. Among the population-based methods, one particular method has been selected: the Particle Swarm Optimization (PSO). PSO was first introduced by Kennedy and Eberhart in 1995 [21] as a development of the Evolutionary Algorithms (EA). The most famous and commonly used algorithm belonging to this family is the Genetic Algorithm (GA). This field of study was first initiated in 1975 by Holland [22] and due to the promising results it was successfully addressed towards optimization problems. GAs take their inspiration from the biological evolution of species inside an environment. The latter is defined by the problem itself and the individuals (also called *chromosomes*) represent the candidate solutions. Iteration by iteration, new individuals are generated through *recombination* and *mutation* of previous ones, imitating Darwin's theory of *natural selection*. The individuals with higher fitness have more probability to survive to the next generation. PSO was instead inspired by the movement of birds in flocks [21]. Even though being classified as an evolutionary algorithm at the beginning, its theoretical background along with its great potential gave origin to a new category of algorithms known as Swarm Intelligence (SI). This algorithm was chosen to perform the optimizations in this work of thesis due to its outstanding performances. Differently from GA, PSO does not rely on probability but on social behavior, implementing rules such as neighbor velocity matching and acceleration by distance [23]. In particular, a multi-objective version of PSO, known as Multi-Objective Particle Swarm Optimization (MOPSO) [24] and specially modified to work with discrete variables, will be used since better suited to the optimization of a multi-deployment mission.

1.3.3 Low-thrust trajectory design

Spacecraft trajectories are characterized by six coupled differential equations [25]. In addition to this, the motion of the spacecraft is perturbed by natural disturbances and by its propulsive system. When considering continuous thrust maneuvers, such as the low-trust case, also a seventh uncoupled

differential equation is needed to model the mass variation in time due to propellant consumption.

When perturbations and thrust are considered it is not possible to analytically solve the system of equations apart from really peculiar cases [26][27]. Solving the equation now becomes an optimal control problem. The control law, given by the time history of the thrust, the in-plane and the out-of-plane thrust angles, can be found with different methods belonging to the following three macro-categories: direct methods [28] [29], indirect methods [30] and shape-based methods.

- **Direct methods** have high computational cost since based on the conversion of the continuous optimal control problem into a high parametric one.
- When using **indirect methods**, the problem is converted into a Two-Points Boundary Values Problem (TPBVP) that can be solved with a lower computational cost with respect to direct methods but requiring an initial guess close to the optimal one.
- The **shape-based methods** instead have relatively low computational cost and provide a good sub-optimal solution, making them the most useful when dealing with large-search domains. This solution can later be used as initial guess for a direct or indirect method to find the optimal one.

For these reasons, since many evaluations of the low-thrust transfers will be necessary to define the overall deploying trajectories, a shape-based approach will be used in this work to solve the optimization problem. Some exact shape-based solutions are summarised by Petropoulos and Sims [31]. These analyses can be defined to be shape-based since the shape of the trajectory is assumed a priori and the required thrust is computed a posteriori. However, these solutions only take into account the planar problem and do not give the possibility to exactly fix the boundary conditions of the trajectory. One of the most used shape-based approaches for low-thrust trajectory computation is the one developed by Conway and Wall, first only for the 2D case [3] and then extended to the three-dimensional one [32], which however only works for really small displacements in the third dimension. Nevertheless, when dealing with planetocentric scenarios like the one object of this thesis, the thrusting of the spacecraft occurs over many revolutions around the main planet. This further increases the complexity of the problem, making the solutions found by Conway ineffective. The choice for the algorithm to be used in the optimization process has therefore fallen on the work by Prinetto [33], who developed an analytical shape-based method particularly suitable for multi-revolution low-thrust trajectory design. This shape-based approach is described in detail in section 4.1.1.

1.4 Thesis outline

This work of thesis will be organized in the following structure:

- First, in chapter 2 the main structure of the routing algorithm will be presented. This algorithm will be first kept as general as possible and applies to any routing problem. Afterward, the approach through which the routing and transfers will be optimized is presented in detail.
- In chapter 3 the routing algorithm will be specialized for the multi-deployment scenario with impulsive maneuvers. Also, in this chapter, a validation of the routing algorithm is performed and commented.
- In chapter 4 the routing algorithm is applied to the low-thrust multi-deployment scenario. Afterward, the low-thrust transfer strategy is compared to the impulsive one.
- Later, in chapter 5 some applications of the multi-deployment algorithm are dealt with. Both constellation and multi-satellites deployment scenarios are taken into consideration
- Finally, chapter 6 critically comments the results achieved in the previous chapters and suggests some possible future works on the matter.

2. Routing architecture development and optimization

This chapter is divided into two main sections:

- **Routing algorithm**: the workflow of the algorithm is explained, still independently from the propulsion of the vehicle at this point. This algorithm will be applied to a multi-deployment mission scenario but is a generic algorithm solving the routing which can be applied to any other routing problem, such as MGA, on-orbit servicing or ADR.
- **Optimization**: the peculiar optimization approach developed and adopted for the optimization of the routing and transfers is presented.

2.1 Routing algorithm

First, the main structure of the algorithm is presented. Afterward, more details about the reasoning behind the developing choices are reported.

2.1.1 Algorithm workflow

The main workflow of the algorithm is shown in Figure 2.1.

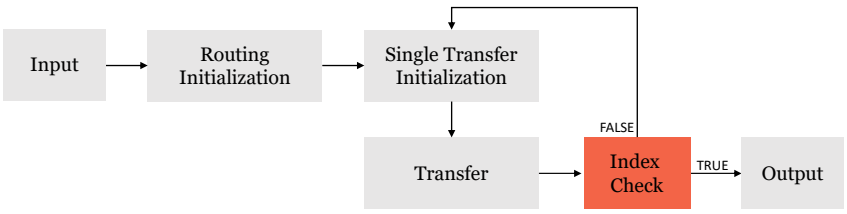


Figure 2.1: Routing algorithm main workflow.

- **Input**: at first, the main inputs are given to the function. The main inputs of the algorithm are the initial wet mass of the vehicle M_0, the

initial time JD_0, the specific impulse I_{sp}, the initial state KP_0 and the two matrices **K** and **P**. The former is a matrix of dimensions $6 \times N$, where N is the number of orbits onto which to deploy the satellites. Each column of the matrix contains the KP of each orbit. Matrix **P** is presented in the next item.

- **Routing Initialization**: In this block, the releasing order is defined. Given N orbits onto which to release the satellites, the optimal path will be one of the possible permutations of the vector $[1, 2, ..., N]$, where each number identifies one of the columns of **K**. The matrix **P** is built outside the algorithm through the operator in eq. (2.1), where $x = [1, 2, 3, .., N-1, N]$.

$$\mathbf{P} = \texttt{perms}(x) \tag{2.1}$$

  The operator builds the matrix **P** such to contain all the possible permutations of x and its size will be $N \times N!$. For instance, if N = 3 the following matrix is created:

$$\mathbf{P} = \begin{bmatrix} 3 & 3 & 2 & 2 & 1 & 1 \\ 2 & 1 & 3 & 1 & 3 & 2 \\ 1 & 2 & 1 & 3 & 2 & 3 \end{bmatrix} \tag{2.2}$$

  Once the matrix **P** is built, only one discrete variable identifying one of the columns of **P** is enough to define the releasing order. The selected column will be referred to as p. Arithmetic overflow might occur when building the matrix **P**. Such issue and more details about the choice of introducing **P** to solve the VRP are discussed in section 2.1.2.

- **Single Transfer Initialization**: Once the releasing order is fixed, the first of the N transfers must be initialized. In this block, the initial mass and date of the transfer are updated, together with the target state position which also depends on the epoch (see section 2.1.3).

- **Transfer**: In this block, the transfer between the current state and the next one in the releasing order is computed.

- **Index Check**: In this block a check about the progress of the releasing mission takes place. If the last released satellite does not correspond with the last element of p, the next transfer is initialized. Otherwise, the mission is completed and the final output can be computed.

- **Output**: Once the last element of p has been reached, the total propellant consumption and duration of the transfer can be computed. In particular, the total values are the sum of the partial contributions of each transfer.

2.1.2 VRP solution

As aforementioned, the problem belongs to the MINLP family, presenting both discrete and continuous variables. The latter are the ones optimizing the transfers, while the discrete variables are needed to define the route of the releasing vehicle. Since the multi-deployment algorithm must run inside an optimizer in order to find the optimal or near-optimal solutions, it was necessary to find a way to help the optimizer to efficiently evaluate different releasing orders. To solve the problem, two main ways were identified:

1. One way to approach the problem may have been working with N discrete variables, each one assuming a value from one to N, which put together in a vector would compose the visiting order of the hopping trajectory. However, this solution did not result to be efficient since a constraint of mutual diversity of the N variables would have to be introduced. Every visiting order vector with two repeating numbers would indeed represent a solution with no practical meaning since it would imply reaching twice the same orbit.

2. In order to speed up the algorithm and guarantee convergence onto a solution with practical meaning, the introduction of the matrix $\mathbf{P}$ was considered. This approach guaranteed that the optimizer only evaluates solutions with practical meaning, i.e. only solutions with no repetitions in the releasing order. Also, this solution resulted to be particularly efficient since thanks to this choice only one discrete variable identifying the column of the matrix in eq. (2.2) is sufficient to define the visiting order of the orbits, allowing to drastically reduce the search space. This variable will be the only discrete variable of the optimization and shall adopt values from 1 to $N!$, which is the number of columns of matrix $\mathbf{P}$.

On the one hand, the latter solution speeds up the convergence of the algorithm with respect to considering N discrete variables to define the visiting order. On the other hand, the building of $\mathbf{P}$ can require large storage space with increasing values of N. For these reasons the matrix $\mathbf{P}$ is built only once outside the multi-deployment algorithm and this approach proves ineffective for values of N larger than 11 (e.g. the storage of matrix $\mathbf{P}$ for $N = 12$ would require 42.8 GB of space). To avoid this limitation, a hybrid optimization approach between a branch and bound method and a heuristic method was implemented, explained in section 2.2.2.

2.1.3 Target state dependence on epoch

Sometimes environmental perturbations are so high that it would be infeasible and even useless counteracting them for the whole duration of the mission. This is the case of the Earth's gravitational perturbation in Low

Earth Orbit (LEO) region. The most important effect of such perturbation is a secular variation of the RAAN of a spacecraft over time. The zonal harmonic J_2 is the only one taken into account in this analysis since it constitutes the main contribution to the gravitational perturbation of the Earth. The rate of change is described by eq. (2.3), where n represents the mean angular velocity on the orbit, $R_\oplus$ the radius of the central attractor, i the inclination of the orbit and p its semi-latus rectum. It is clear that the closer to the Earth's surface, the higher the rate of change.

$$\dot{\Omega}_{sec} = -\frac{3nR_\oplus^2 J_2}{2p^2} cos(i) \tag{2.3}$$

Since changing the RAAN basically is a plane change maneuver, the latter is really expensive from a propellant consumption point of view and must therefore be avoided as much as possible. However, most times satellites let the RAAN shift according to the J_2 accelerations while compensating only some short and long period variations due to the J_2 effect, such as the one of the semi-major axis. Two are the most important cases in which the RAAN is left free to shift:

1. For instance, when dealing with a **constellation** of satellites usually what really is important is the RAAN difference of the orbital planes rather than the absolute values of this parameter. Orbital planes are left free to shift their RAAN, as long as the shift is the same for all of them.

2. Another case in which the RAAN is willingly left to shift is when a satellite is placed on Sun-Synchronous Orbit (**SSO**). These orbits are characterized by a net effect of the Earth gravitational perturbation which makes the RAAN precesses in such a way that a satellite on this orbit passes over any point of the Earth surface at the same local time, meaning that the angle between the Sun and the orbital plane stays constant. When describing a SSO, the RAAN is substituted by the Local Time of the Ascending Node (LTAN). LTAN represents the time when the satellite crosses the equator when traveling from the south pole to the north pole, that is why it is called *ascending*. Consequently, it defines the time of day at that Earth location when the satellite is overhead. A SSO is characterized by a constant LTAN.

These are the reasons why the target state depends on the epoch and this dependence must be accounted for in the multi-deployment algorithm. In the block **Single Transfer Initialization**, once the JD is updated it is possible to accordingly update the target state through eq. (2.3), knowing the initial condition which is stored in matrix **K**. By doing so only the secular variation of the orbital parameters is taken into consideration. However, integrating

the target state to consider each oscillation would not only dramatically slow down the algorithm but also lead to no improvement in terms of propellant and time estimation.

2.2 Optimization

Once the routing algorithm is built, this algorithm is run through an optimizer to find the optimal or near-optimal visiting order and transfer strategy. In this section, first, an overview of the kind of optimization adopted is given. Afterward, the particular optimization approach developed to solve the problem is presented. Finally, the way the constraints are introduced into the optimization process is explained.

2.2.1 Multi-objective optimization

When dealing with a multi-deployment mission, two are the most important values to be careful about: the overall propellant consumption and the mission duration. For this reason, a bi-objective optimization aimed at minimizing these two values was chosen.

While the result of a single-objective optimization problem is a single solution, in a non-trivial multi-objective optimization problem the result is a set of optimal solutions. This is due to the fact that there is no single solution that simultaneously minimizes all the objectives. The set of optimal solutions is composed of all the non-dominated solutions found by the optimization method, also known as Pareto solutions. A solution S1 dominates a solution S2 if each of the objectives of the solutions S1 are most performing than the ones of S2. In the case of bi-objective optimization, the set of non-dominated solutions presents itself as a line known as Pareto front. The classic output of bi-objective optimization is represented in Figure 2.2. The non-dominated solutions compose the set of optimal solutions found by the algorithm.

Alternatively, a single-objective optimization could be performed but a constraint on the other quantity must be introduced. This would not be effective for two reasons:

1. It is not always easy or possible to define a net constraint of a quantity. Also, introducing the latter to one of the two quantities, would automatically discard solutions with a slightly higher value than the constraint but maybe with a really convenient final objective value.

2. The final solution of the single-objective optimization would be very likely to have the constrained quantity adopting a value very close to the constraint. This would not allow the investigation of solutions with a lower value of the constrained quantity but with a slightly worse objective value.

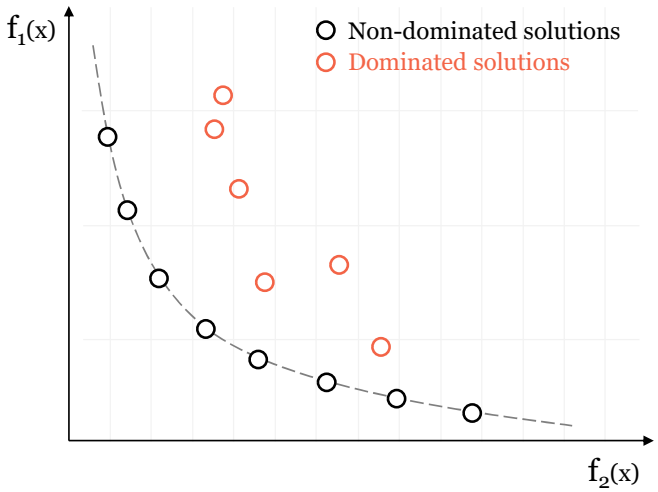


Figure 2.2: Pareto front.

For these reasons, a bi-objective optimization was adopted to solve the problem. As already stated in section 1.3.2, due to the vastness of the search space it would be infeasible to choose an exact method for the optimization. It was necessary to select an optimization method that trades the optimality of the solution for computational speed. For this reason, a heuristic method was selected, in particular a population-based method which is the most suitable for very large search space problems, since based on the simultaneous evaluations of more solutions. From now on, the MOPSO [24] method, specially modified to work with discrete variables, will be used to perform all the multi-objective optimizations.

2.2.2 Optimization approach

While the optimization method selected was the MOPSO, a particular approach to deal with the optimization of the problem was developed and is presented in this section.

On the one hand, using the matrix **P** proved extremely effective in the numerical validation of the routing algorithm (see section 3.4). On the other hand, it does not allow to deal with a problem with more than 11 orbits to reach. To overcome this limitation a branch and bound based heuristic approach was considered. This approach is a hybrid between the classical exact branch and bound method and a heuristic algorithm. Given a set of N orbits to reach onto which to release the satellites, instead of dealing with the whole hopping trajectory at once, the problem is broken up into smaller and consequential subproblems. A parameter r which defines the dimension of the subproblem is chosen, defining also the number of iterations necessary to solve the problem of reaching the N orbits. The number of iterations

N_{iter} needed to solve the problem can be computed by eq. (2.4), where the operator rounds the result of the fraction to the nearest integer greater than or equal to it.

$$N_{iter} = \texttt{ceil}\left(\frac{N}{r}\right) \tag{2.4}$$

In the case in which the remainder after the division of N by r is different from zero, the last iteration is performed with a subset of size r equal to the remainder. For instance in the case of $N = 14$ and $r = 4$, four iterations will be needed, the latter of which with $r = 2$.

When adopting this hybrid approach, the matrix $\mathbf{P}$ to give as input to the routing algorithm is built through the use of a different operator than eq. (2.1). The new operator is shown in eq. (2.5) and builds a matrix which, given a set of N orbits to reach, only contains the permutations of a subset of r elements of the vector x.

$$\mathbf{P} = \texttt{subperms}(x, r) \tag{2.5}$$

The number of possible permutations N_{perms}, in this case, can be found through eq. (2.6).

$$N_{perms} = \frac{N!}{(N-r)!} \tag{2.6}$$

The size of the matrix $\mathbf{P}$ will be $r \times N_{perms}$. For instance, with N = 4 and $r = 2$, matrix $\mathbf{P}$ coming as output from eq. (2.5) would appear as in eq. (2.7).

$$\mathbf{P} = \begin{bmatrix} 2 & 1 & 3 & 1 & 4 & 1 & 3 & 2 & 4 & 2 & 4 & 3 \\ 1 & 2 & 1 & 3 & 1 & 4 & 2 & 3 & 2 & 4 & 3 & 4 \end{bmatrix} \tag{2.7}$$

However, again, the value of r must be carefully selected to avoid the memory problem which occurs computing the permutations of the whole set. It can be easily seen from eq. (2.6) that the larger the subset, the higher the number of columns of the matrix $\mathbf{P}$ and therefore the higher its storage space required. Choosing therefore a too high parameter r defining the dimension of the subsearch would not allow solving large-N problems anyway.

Optimization approach pseudo code

The pseudo code of the optimization approach is presented in Alg. 1, whose steps are described below, and is represented by the scheme in Figure 2.3. In the latter, each circle represents a single solution. The grey solutions actually represent several solutions, whose number is unknown a priori due to the heuristic nature of the optimization.

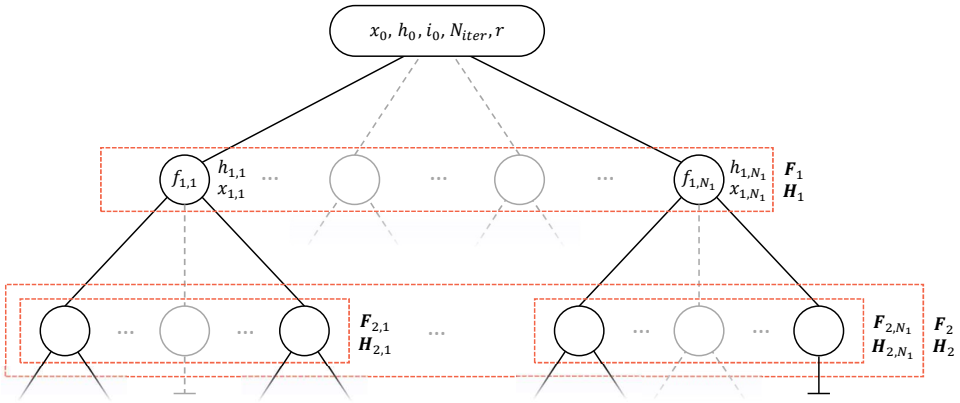


Figure 2.3: Branch and bound based heuristic approach scheme.

Algorithm 1: Branch and bound based heuristic approach

Data: x_0, i_0, h_0, N_{iter}, r
Result: $\mathbf{F}_{N_{iter}}$, $\mathbf{H}_{N_{iter}}$

begin
  $[\mathbf{F}_1, \mathbf{H}_1] = \texttt{branching}(x_0, i_0, h_0, r)$
  for $j = 2$ **to** N_{iter} **do**
    $N_{j-1} = \texttt{length}(\mathbf{H}_{j-1}(:, 1))$
    for $k = 1$ **to** N_{j-1} **do**
      $h_{j-1,k} = \mathbf{H}_{j-1}(k, :))$
      $x_{j-1,k} = \texttt{exclude}(x_0, h_{j-1,k})$
      $i_{j-1,k} = \mathbf{F}_{j-1}(k, :))$
      $[\mathbf{F}_{j,k}, \mathbf{H}_{j,k}] = \texttt{branching}(x_{j-1,k}, i_{j-1,k}, h_{j-1,k}, r)$
    end
    $\mathbf{H}_j = [\mathbf{H}_{j,1}; \mathbf{H}_{j,2}; \ldots ; \mathbf{H}_{j,N_{j-1}}]$
    $\mathbf{F}_j = [\mathbf{F}_{j,1}; \mathbf{F}_{j,2}; \ldots ; \mathbf{F}_{j,N_{j-1}}]$
    $[\mathbf{F}_j, \mathbf{H}_j] = \texttt{bounding}(\mathbf{F}_j, \mathbf{H}_j)$
  end
  $[\mathbf{F}_j, \mathbf{H}_j] = \texttt{boundingpareto}(\mathbf{F}_j, \mathbf{H}_j)$
end

- The inputs to the algorithm are the vector of all the elements still to be reached x, the vector of initial conditions i, the vector containing the indices of the destinations already reached h. At the first iteration $x_0 = [1, 2, .., N]$ and h_0 is initialized as an empty vector. In addition to these, the number of iterations N_{iter} computed in eq. (2.4) and the sub-search dimension r are also given as input to the algorithm.

- The first iteration is performed outside the **for** loop. The operator `branching`, starting from one initial condition, runs the heuristic optimizer and finds a set of partial solutions which make the first branches of the algorithm. The matrix $\mathbf{P}$ at this iteration is computed through eq. (2.5), with x_0 as input. The heuristic algorithm only gives as output the solution which respect the bounding criteria (explained in the next paragraphs). Each solution is characterised by $f_{j,1}$, $h_{j,1}$ and $x_{j,1}$ vectors. The survived solutions are stored in two matrices:
  - $\mathbf{F}_j = [f_{j,1};\ f_{j,2};\ \ldots\ ;\ f_{j,N_j}]$ is the matrix containing the final conditions of each solution, including the values of the objectives.
  - Each row of $\mathbf{H}_j = [h_{j,1};\ h_{j,2};\ \ldots\ ;\ h_{j,N_j}]$ contains the indices of the elements reached by the respective partial solution.
- Now the first **for** loop begins. At each iteration, N_{j-1} is computed, which is the number of the branches coming from the previous iteration. The operator `length` computes the number of elements of the vector of input. Each of the branches represents a solution that must be expanded in the following **for** loop.
- One branch at a time, the iteration is initialized defining $h_{j-1,k}$, $i_{j-1,k}$ and $x_{j-1,k}$. The latter in particular is defined by the operator `exclude`. This operator cancels from x_0 all the elements already reached by that partial solution, identified by $h_{j-1,k}$. The new vector $x_{j-1,k}$ will have dimension $N - r \cdot j$.
- From each branch, a new set of branches is found again through the operator `branching`. At each iteration a different $\mathbf{P}$ is given as input to the routing algorithm, again computed through eq. (2.5), each time with $x_{j-1,k}$ as input. The branches are stored in $\mathbf{F}_{j,k}$ and $\mathbf{H}_{j,k}$.
- Once all the branches have been expanded, the solutions are stored in the matrices $\mathbf{F}_j$ and $\mathbf{H}_j$, which include all the branches born from the current iteration.
- Before starting the next iteration, the bounding criteria must be applied to $\mathbf{F}_j$ and $\mathbf{H}_j$. While it is true that each $\mathbf{F}_{j,k}$ is composed by solutions which survived the bounding inside the single optimization, now they must be compared to the all the other set of solutions of the iteration. The bounding criteria are applied by the operator `bounding` and the new $\mathbf{F}_j$ and $\mathbf{H}_j$, containing only the survived solutions, are given as output. In Figure 2.3 the solutions which do not survive the bounding criteria are indicated with the symbol $\perp$.
- The solutions of the iteration $j-1$ which survived the `bounding` operator, are now expanded themselves.

- After the last iteration, if necessary, a more stringent bounding criterion is applied to the final set of solutions. In particular, through the use of the operator `boundingpareto` only the non-dominated solutions are kept.

Bounding criteria

The operator `bounding`, given a set of solutions and some bounding criteria, only keeps the solutions that respect the latter while discarding all the others. For a single-objective optimization, the criterion could be discarding all the solutions out of a certain percentage range from the optimal solution found. For a multi-objective solution, the choice of a bounding criterion is more complex since usually a set of optimal solutions is found. One way might be discarding all the solutions not belonging to the Pareto front of the sub-problem, whose properties have been explained in section 2.2.1. However, there is the risk of discarding some partial solutions really close to the Pareto front, which might eventually become optimal solutions when extending them. Therefore, also in the case of a multi-objective optimization, a region of solutions must be selected: all the solutions belonging to the Pareto front or within a certain percentage range from one of the solutions in the front can be kept and then extended. These solutions outside the Pareto front but which survive the bounding criterion are referred to as "partially dominated solutions", even though they are actually fully dominated according to the definition of dominated solutions in section 2.2.1. The value r_{perc}, chosen between 0 and 1, sets the percentage range within which a dominated solution is considered to be only partially dominated. A schematic representation of such solutions is reported in Figure 2.4. Indicating with $f_{1,D}$ and

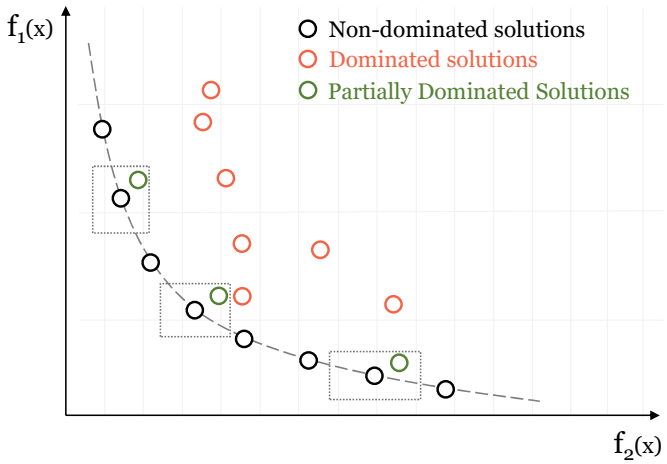


Figure 2.4: Pareto front and partially dominated solutions.

$f_{2,D}$ the objectives of a dominated solutions and with $f_{1,P}$ and $f_{2,P}$ the ones

of the closest non-dominated solution, a dominated solutions is considered to be partially dominated if both the criteria in eq. (2.8) are met.

$$\frac{f_{1,D} - f_{1,P}}{f_{1,P}} < r_{perc} \tag{2.8a}$$

$$\frac{f_{2,D} - f_{2,P}}{f_{2,P}} < r_{perc} \tag{2.8b}$$

Graphically, each Pareto solution has a rectangle with sides of length $r_{perc} \cdot f_{1,P}$ and $r_{perc} \cdot f_{2,P}$ which defines the range into which a dominated solution is considered to be partially dominated.

2.2.3 Constraints introduction

When dealing with engineering problems, it is of paramount importance that an algorithm allows the introduction of constraints to some of the variables. In the routing algorithm, the constraints will be introduced by the use of penalty functions.

Penalty functions allow treating of constrained problems as unconstrained problems, introducing an artificial penalty when the constraint is violated. The constraints can be applied both to a variable or to one or more of the objectives. However, the introduction of penalty functions may create severe slope changes or discontinuities in the solution space, which could interfere with the optimization algorithms chosen in this thesis. For example, introducing a fixed penalty when a constraint is violated may lead to these kinds of complications since the optimization algorithms investigate the search-space focusing on the directions where the objective functions improve. A constant penalty might interfere with this research and therefore with the convergence of the method since no improvement in the objective function violated would be found and it would be more difficult to redirect the search towards regions that respect the constraint. One way to solve this problem is by introducing a penalty function whose penalty is proportional to the amount of violation of the constraint. By doing so, the algorithm even if in a region of the search space which violates the constraint is able to exit from such region. For instance, considering a generic objective f_1 and an upper limit L_1, the objective function is modified adding the quadratic loss function $\lambda(f_1, L_1)$ in eq. (2.9).

$$\lambda(f_1, L_1) = \max(0, f_1 - L_1)^2 \tag{2.9}$$

The new formulation of the objective is presented in eq. (2.10),

$$f_1 = f_1 + F \cdot \lambda(f_1, L_1) \tag{2.10}$$

where F is the penalty factor, a scalar greater than 0 and arbitrarily chosen depending on the order of magnitude of f_1.

3. Multi-deployment impulsive algorithm

Once the main structure of the routing algorithm has been presented, it is possible to go in-depth with impulsive and low-thrust versions of the multi-deployment algorithm. This chapter is organized in the following way:

- **Single transfer**: the first section briefly presents how the impulsive single transfers are computed.
- **Transfer strategy**: the second section explains the strategy selected to perform the transfers between the states of matrix **K**.
- **Multi-insertion on same orbit**: this section explains the algorithm developed to take into account the scenario in which more than one satellite has to be released on the same orbit but at different anomalies.
- **Numerical results**: in the last section the impulsive routing algorithm is tested on a problem already solved in literature, to test the performances of the main algorithm structure presented in the previous chapter.

Once the impulsive transfer strategy is chosen, it is enough to integrate it with the routing algorithm of Figure 2.1 to create the impulsive multi-deployment algorithm represented in Figure 3.1.

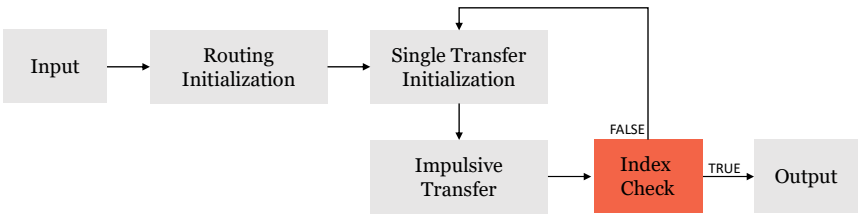


Figure 3.1: Impulsive multi-deployment algorithm algorithm main workflow.

3.1 Single transfer

The single transfers, when dealing with impulsive maneuvers, are simply modeled as a solution of the Lambert problem [34]. Once the transfer trajectory is defined, the cost of the transfer can be easily obtained by computing the differences in the velocity vectors at the injection onto the transfer orbit and at the moment of the arrival at the desired state. From the solution of the Lambert problem, the transfer orbit is defined and it is, therefore, possible to compute the magnitude of the Δv of the two impulses. In this case, the only variable of the transfer optimization is the Time Of Flight (TOF) of the transfer.

Usually, when dealing with impulsive maneuvers the cost of the transfer is measured in the total change of velocity Δv and it is therefore considered the objective of the minimization in an optimization problem. The issue with adopting such objective is that the purpose to save propellant mass coincides with the minimization of the Δv only under the assumption that the mass variation of the vehicle occurs solely by depletion of the fuel. This is not true for the application dealt with in this thesis, since after each transfer a satellite is deployed varying the dry mass of the vehicle. In that case, the minimization of the propellant mass must be addressed directly. After each transfer the Δv cost must be converted into propellant cost through the Tsiolkovsky equation in eq. (3.1), where g_0 is the average Earth gravity acceleration, M_0 the mass of the spacecraft before the transfer and M_f the mass at the end.

$$\Delta v = \mathrm{I}_{sp} \cdot g_0 \cdot \ln\left(\frac{\mathrm{M}_0}{\mathrm{M}_f}\right) \tag{3.1}$$

From eq. (3.1), M_f can be obtained and the propellant cost of the transfer $\mathrm{M}_{\mathrm{prop}}$ is found simply by computing the difference between initial and final mass.

In spite of its simplicity, this transfer strategy is not efficient at all when connecting two orbits with large differences in RAAN. Since having in matrix **K** orbits characterized by different values in RAAN was considered a really likely case in a multi-deployment mission scenario, a different strategy was chosen to perform the transfers.

3.2 J_2 exploiting transfer strategy

To perform the transfers in the most efficient way, a transfer strategy exploiting the secular effect of the J2 perturbation was planned and is explained in the next paragraphs. The main idea behind this different approach is to change the RAAN of the spacecraft not by thrusting the spacecraft but only exploiting the gravitational perturbation due to the not spherically symmetric mass distribution of the central attractor [5]. The RAAN of a spacecraft

on a given orbit can be changed for free by waiting the necessary amount of time without counteracting the J2 perturbation. By doing so, the rate of change of the RAAN would be the one in eq. (2.3). Alternatively, the spacecraft can also move to another orbit in order to make the desired change of RAAN happen faster at a cost of a little propellant consumption. This two-legs transfer option is the most efficient in terms of propellant consumption and turns the transfer problem into an optimization problem whose variables are the KP of the intermediate orbit onto which to stationary for the change of RAAN and the TOF of the two transfer legs. The workflow of the transfer strategy is shown in Figure 3.2

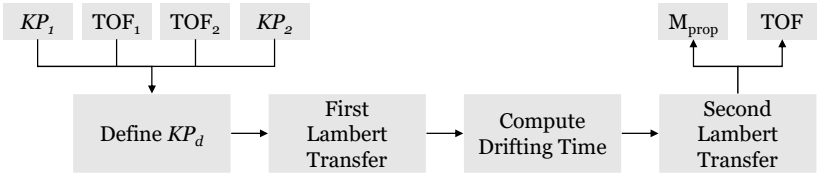


Figure 3.2: J2-exploiting impulsive transfer strategy workflow.

- **Inputs**: The starting orbit KP_1 and the target orbit KP_2 are the two main inputs to the function. TOF_1 and TOF_2 are the TOF of the two legs of the transfer strategy and are the variables of the optimization for each single transfer.
- **Drifting Orbit Definition**: Starting from the two orbits, it is possible to define the orbit onto which to stationary to shift the RAAN, here referred to as drifting orbit KP_d. The drifting orbit is completely defined by 6 variables, which are its 6 KP. The way such parameters are defined is presented in Table 3.1.

a_d	e_d	i_d	Ω_d	ω_d	θ_d
var	0	var	Ω_1	$\frac{1}{2}(\omega_1 + \omega_2)$	$\theta 1 + pi$

Table 3.1: KP of the drifting orbit for impulsive maneuvers.

As it can be seen from eq. (2.3), the three parameters which affect the RAAN variation are the semi-major axis a, the eccentricity e and the inclination i. For this reason, not all the KP are considered as variables in order to reduce the size of the optimization problem:

  - The semi-major axis is the one which has the greatest impact on the RAAN variation and it is also an element whose value is relatively cheap to control and change, making it the main variable of the optimization.

- Eccentricity affects the RAAN change since a high value means an orbit perigee at a lower altitude and therefore more subject to the gravitational asymmetries of the central planet. However, it is preferred to have a uniform influence of such asymmetry on the spacecraft over the revolutions and therefore this value is chosen to be 0. This parameter might be taken as a variable for further and more detailed optimizations; it may be more relevant when dealing with starting or target orbits with high eccentricity.
- The inclination is a parameter that also greatly affects the RAAN variation but it is also extremely expensive to change. Therefore, it is considered as variable but only when a change of inclination is necessary from original to target orbit. In that case, the value of the inclination of the drifting orbit is selected between i_1 and i_2. In the case in which i_1 and i_2 have the same value, the inclination is kept the same as the ones of the two orbits unless polar orbits are dealt with. Polar orbits do not experience the RAAN secular variation and, for this reason, it is necessary to depart from 90° inclination to shift the orbital plane.
- The argument of perigee ω of an orbit does not affect the secular variation of the RAAN. For this reason, ω_d was not taken as variable of the problem and was arbitrarily set to have a halfway value between ω_1 and ω_2.
- The RAAN of the drifting orbit is chosen to be equal to Ω_1 since the RAAN variation will be fully achieved through the exploitation of the secular effect of the perturbation.
- The true anomaly θ was also discarded as variable since it does not affect the secular RAAN variation and therefore was considered to be $\theta 2 = \theta 1 + pi$ to resemble as much as possible a Hohmann transfer.

- **First transfer**: Once the drifting orbit is defined, the cost and duration of the first transfer are computed.

- **Drifting Time Computation**: Once the target orbit has been reached, it is necessary to compute the amount of time necessary to close the RAAN gap. Generally speaking, knowing the elements of the drifting orbit KP_d and the ones of the target orbit KP_2 it is possible to compute the RAAN gap (eq. (3.2a)) at the moment of the arrival on the drifting orbit and the relative drift rate (eq. (3.2b)). Once these two quantities are known, it is possible to compute the waiting time (eq. (3.2c)) necessary to have $\Omega_2 = \Omega_d$, at a value different from the

two original ones.

$$\Delta\Omega = \Omega_2 - \Omega_d \tag{3.2a}$$

$$\Delta\dot{\Omega} = \dot{\Omega}_d - \dot{\Omega}_2 \tag{3.2b}$$

$$t_{wait} = \frac{\Delta\Omega}{\Delta\dot{\Omega}} \tag{3.2c}$$

- **Second Transfer**: Once the RAAN has been updated for both the orbits after the stationing onto the drifting orbit, the second transfer to the target orbit KP_2 can effectively be computed.
- **Outputs**: At the end, the whole cost of the two-legs transfer is computed. In terms of propellant, the total amount needed to reach the target orbit is the sum of the propellant consumed in the two legs of the transfer. In terms of time, the total duration of the transfer is the sum of the TOF of the two legs and also the waiting time onto the drifting orbit.

Numerical example

It is interesting to find some numerical results about the J_2 exploiting strategy. The KP of the two starting and target orbits are the ones in Table 3.2. The spacecraft initial mass is considered to be 100 kg with a specific impulse I_{sp} of 350 s.

	a [DU]	e [-]	i [deg]	Ω [deg]	ω [deg]	θ [deg]
KP_1	1.1	0.1	67	30	0	0
KP_2	1.3	0	67	0	0	0

Table 3.2: KP of starting and target orbit for impulsive transfer strategy example.

The multi-objective optimization was performed through the MOPSO, whose two main parameters to tune are the population size N_p and the maximum number of generations M_{gen}. For the optimization of the single transfer these parameters were chosen to assume the values $N_p = 50$ and $M_{gen} = 100$. The variables of the optimization, as already stated in Table 3.1, were the semi-major axis, the inclination of the drifting orbit and the TOF of the two transfers. The lower and upper bounds, lb and ub, of the optimization are shown in Table 3.3. The upper bound of the TOF of the

Parameter	lb	ub
a_d [DU]	1.05	1.5
i_d $[deg]$	67	67
TOF_1 [d]	0	T_2
TOF_2 [d]	0	T_2

Table 3.3: Lower and upper bounds for the impulsive transfer strategy example.

two transfers was set equal to the larger period of the two orbits T_2. Since the inclination of starting and target orbit are the same, fictitious lower and upper bounds for this variable were considered. Performing the optimization with a range of inclination as solution space would lead to the same results in the optimization of the single transfer. However, when dealing with a multi-deployment scenario with N different transfers, considering a range of inclination for each transfer would critically increase the size of the solution space negatively affecting the quality of the final solution. For this reason, the range of inclination can be neglected if the orbits belong to the same inclination plane.

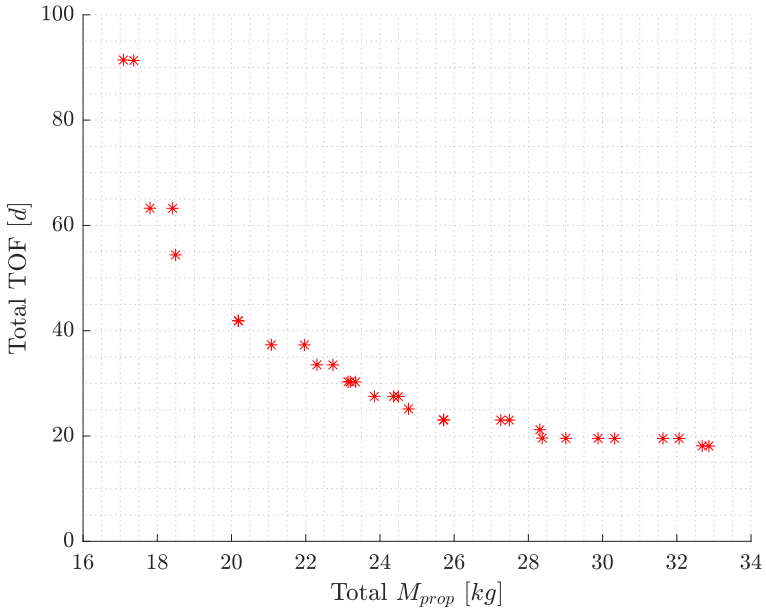


Figure 3.3: J2-exploiting impulsive transfer strategy example results ($M_0 = 100\ kg$; $I_{sp} = 350\ s$).

The results of the bi-objective optimization are shown in Figure 3.3. When dealing with impulsive maneuvers, the times of flight of the transfers are short and the greatest part of the duration of the mission is the drifting time. However, as it can be seen from Figure 3.3, adopting the J_2 exploiting transfer strategy with impulsive maneuvers does not necessarily lead to a short duration of the mission. These results will be later compared to the low-thrust case in section 4.2.

3.3 Multi-insertion on same orbit

When more than one satellite has to be released onto the same orbit, a phasing maneuver must be performed to release them at the correct true anomaly. When not dealing with low-thrust, a phasing maneuver is a two-impulse Hohmann transfer from and back to the same orbit [35]. The first impulse is given at the perigee of the orbit in order to insert on one with a smaller (larger) period when the target is ahead (behind) the chaser in terms of anomaly. The period T_2 of the intermediate orbit is computed with eq. (3.3), where T_1 is the period of the original orbit, t_{AB} the time needed on the original orbit to close the anomaly gap and N_{rev} the number of revolutions that needs to wait for on the intermediate orbit. Of course the higher N_{rev}, the longer the duration of the phasing maneuver, the lower the cost since T_2 will be more similar to T_1 and therefore requiring a smaller Δv. Once waited for the desired number of revolutions, the chaser will exactly rendezvous the target at the perigee and a second impulse is given to insert again on the original orbit.

$$T_2 = T_1 - \frac{t_{AB}}{N_{rev}} \tag{3.3}$$

The way the phasing maneuver will be taken into account is presented in Figure 3.4. The value of N_{rev} is estimated iteratively, starting from $N_{rev} = 1$,

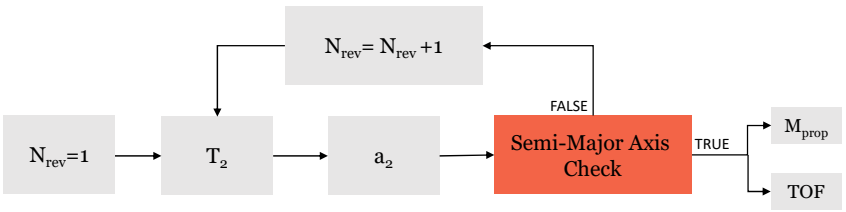


Figure 3.4: Phasing maneuver algorithm.

such that a_2 is greater than a certain fraction of a_1, where a_1 refers to the original orbit and a_2 to the orbit onto which to wait. By arbitrarily fixing a low value of N_{rev}, such as 1, the algorithm would eventually select an a_2 extremely different than a_1 which would require a really large propellant

consumption. For this reason, a fraction of a_1 which defines the lower bound of a_2 must be chosen, depending on the orders of magnitude of the semi-major axes. Whether a_2 is higher or not the lower bound is checked in the **Semi-Major Axis Check** block. Once the period T_2 of the orbit onto which to wait for N_{rev} revolutions is known, it is possible to compute its semi-major axis and subsequently the cost Δv_H of the Hohmann transfer to reach it and to go back to the original orbit. The propellant mass necessary is then computed once Δv_H is known and inverting the Tsiolkovsky equation in eq. (3.1).

3.4 Numerical results

It was interesting to compare the results obtained applying the impulsive algorithm to problems whose solution was known in order to validate it and also to assess the algorithm quality and efficiency. The algorithm validation was carried on by solving the problem faced in the 5^{th} GTOC, whose solution is known and was already used for validation in several papers [4] [36]. The scenario of this problem is different from the multi-deployment mission which is the main focus of the algorithm developed in this thesis. However, the two problems share similar features and therefore by applying only really little changes to the algorithm it was possible to apply it to this different problem. The problem deals with three versions of a multiple asteroids rendezvous task, respectively with 4, 8 and 16 targets. The optimization approach developed in section 2.2.2 to enlarge the capabilities of the algorithm only suits multi-objective optimization (due to the branching and bounding criteria chosen). For this reason, the algorithm will only be tested on the 4 and 8 target cases.

The workflow of the algorithm stays the one of the routing algorithm represented in Figure 2.1. In this case, the impulsive transfer will be performed as a single Lambert transfer, the first transfer option reported in section 3.1. Also, besides the TOFs of the transfers, other variables for the optimization had to be introduced: the departure date t_0 from Earth (which influences the state of the Earth at that date) and the exploration time t_{exp} on each asteroid (which is the time passing from the arrival at one asteroid to the departure from it). These two kinds of variables were not considered in the multi-deployment algorithm but can easily be added if necessary. The constraints on the lower and upper bounds of the variables were fixed and are listed in Table 3.4. The date of departure from Earth is expressed in Modified Julian Date (MJD).

The starting point of the hopping path is the Earth, whose state at the epoch 54000 Julian Date (JD) is reported with the ID 0 in Table 3.5. The other IDs represent the states of the eight targets of the multiple rendezvous trajectory at the epoch 2456600.5 JD.

Parameter	lb	ub
TOF_1 [d]	0	730
t_{exp} [d]	7	365
t_0 [MJD]	57023	61041

Table 3.4: 5^{th} GTOC problem lower and upper bounds.

ID	a [AU]	e [-]	i [deg]	Ω [deg]	ω [deg]	M [deg]
0	0.9999880	0.0167168	0.00089	287.61578	175.40648	257.60684
1	1.0377497	0.0740190	1.27980	196.84658	111.32136	220.55456
2	1.0537339	0.0604714	0.23523	216.10320	134.34440	242.72351
3	1.0099643	0.0830940	1.43929	43.48569	273.55713	57.93400
4	0.9593424	0.1448353	0.72205	36.86737	316.29050	146.08995
5	1.0617043	0.0515307	1.26705	253.32360	316.76764	115.45581
6	1.0382537	0.1065636	0.55071	225.80291	281.99557	136.76768
7	1.0332190	0.0684246	0.26337	21.04400	300.95352	345.64862
8	1.0235198	0.0985837	1.40948	139.83156	94.80897	84.23810

Table 3.5: 5^{th} GTOC problem initial Earth and target conditions.

PSO was chosen as computational method for the single-objective optimization to minimize the total Δv of the hopping journey. In particular, the `particleswarm` function of MATLAB Global Optimization Toolbox [37] was used. The default tuning parameters of the algorithm were adopted in this case; a research of the optimal ones might further increase the performances reported in the next paragraphs.

For both the 4 and the 8 targets cases, respectively identified as Case 1 and Case 2, the problem was solved 10 times and the results are shown in Table 3.7. The results of the routing algorithm (M1) here proposed are compared to the ones found by solving it through two different methods (M2 and M3). The three methods are summarized in Table 3.6.

M1	Routing algorithm
M2	GA with search enhancement
M3	Two Phase Algorithm (TPA)

Table 3.6: Methods for routing algorithm validation.

The results are compared to the ones found by Zhang et al [4], who tried different computational methods to solve the problem. Only the method

which provided the best results is here reported, which are the ones found by performing the minimization through the adoption of a GA with search enhancement (M2). The proposed method will be compared also to the one proposed by Bang and Ahn [36] which consists in a TPA (M3) characterized by a first phase in which some elementary solutions are found which are later used as starting point to solve the TSP.

The results are shown in Table 3.7, which confirms the validity of the algorithm and also proves its quality in performances. Only the information about the best results found by adopting M3 was available.

Method	Case 1 [km/s]			Case 2 [km/s]		
	Best	Mean	STD	Best	Mean	STD
M1	6.068	7.558	0.779	17.350	20.989	1.921
M2	6.397	7.307	0.570	19.153	22.978	3.044
M3	6.360	-	-	16.400	-	-

Table 3.7: Results comparison for algorithm validation.

It is interesting to compare not only the results but also the rapidity with which the algorithm finds its final solution. The computation procedure presented in this report ran on a personal computer with an Intel(R) Core(TM) i7-7500 (2.7 GHz) processor and a 16 GB RAM. However, the rapidity of the codes is measured through the evaluation of how many times the Lambert functions are called to find the final solution and therefore independently on the computing machine. For the proposed method M1 the averages of the 10 runs are reported in Table 3.8 and compared to the number of Lambert calls of the other methods [36]. The results indicate that the computational resource spent by the proposed method is about one order of magnitude smaller than adopting M2 and two than the M3, even though the latter was capable to find a better solution for Case 2.

Method	Number of Lambert routine calls	
	Case 1	Case 2
M1	88,000	348,320
M2	960,000	7,680,000
M3	3,526,933	25,392,677

Table 3.8: Number of Lambert routine calls comparison.

All things considered, it is possible to deduce that the algorithm works and finds reliable solutions to the problem. Also, it is competitive with

respect to similar algorithms in terms of solutions found and especially of computational cost.

4. Multi-deployment low-thrust algorithm

After the development of the impulsive multi-deployment algorithm, the low-thrust version was implemented and is presented in this chapter. The latter is organized in the following way:

- **Single transfer**: the first section presents the method with which the low-thrust single transfers are computed and the changes applied to it to apply it to the multi-deployment scenario.
- **Transfer strategy**: the second section explains the strategy selected to perform the transfers between the states of matrix **K**. The performances are compared to the ones of the parallel impulsive transfer strategy.
- **Multi-insertion on same orbit**: this section explains how the phasing maneuvers were taken into account in the low-thrust scenario.
- **Numerical results**: in the last section the low-thrust algorithm is tested on a simple multi-deployment scenario and the hybrid optimization approach presented in section 2.2.2 is critically analyzed.

In the same way, done for the impulsive case, the low-thrust transfer strategy must be integrated with the routing algorithm of Figure 2.1 to create the impulsive multi-deployment algorithm, schematized in Figure 4.1.

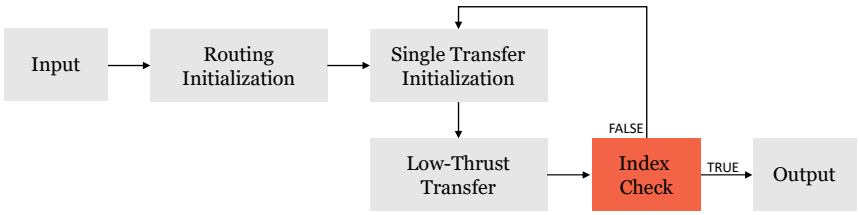


Figure 4.1: Low-thrust multi-deployment algorithm main workflow.

4.1 Single transfer

First, the algorithm chosen for the trajectory computation is presented. Afterward, how the algorithm deals with the anomaly of the target state and its suitability to the deployment scenario is discussed. Finally, the changes applied to shape-based algorithm to apply it to the multi-deployment mission are described.

4.1.1 Shape-based algorithm

A 3-dimensional shape-based algorithm [33] was selected for the single transfers since particularly suited to planetocentric mission scenarios. The shape of the transfer is proposed a priori as a non-linear interpolation of similar and consecutive orbits. At this stage of the trajectory computation, only keplerian dynamic is taken into consideration for the construction of the transfer orbit. Given the initial and final states, the algorithm recovers the trajectory and the dynamics of the transfer. The workflow of the shape-based algorithm is presented in Figure 4.2.

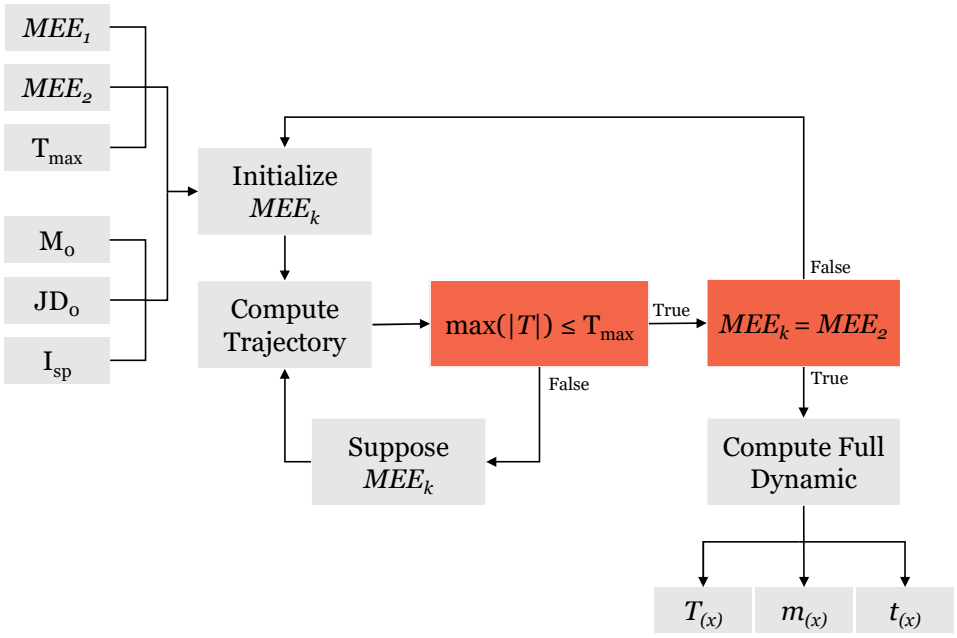


Figure 4.2: Shape-based algorithm workflow.

- **Inputs**: the initial and the final states in Modified Equinoctial Elements (MEE) are given as input, respectively referred to as MEE_1 and MEE_2. The other inputs are the spacecraft maximum thrust available

T_{max}, the specific impulse I_{sp}, the initial mass M_0 and initial time JD_0 (time is expressed in JD).

- **Intermediate Orbit Initialization**: at the first iteration MEE_k is initialized equal to MEE_2.

- **Trajectory computation**: the trajectory is computed. Both the distance from the attractor and the declination above the plane identified by the initial and final states, from here on referred to as reference plane, are parametrized as a function of the non-dimensional angular displacement x in the reference plane.

- **Thrust Check**: The thrust history *T(x)* is recovered and its peak in absolute value is compared to T_{max}. If the thrust excess constraint is not respected, the algorithm introduces one fictitious intermediate orbit located via several numerical procedures, such as the Newton or the bisection methods, aiming to make the thrust peaks coincide with the maximum thrust available of the spacecraft.

- **Intermediate Orbit Check**: Once MEE_k has been reached, whether or not it is equal to the original target state MEE_2 is checked. If not, a new intermediate MEE_k is introduced and the cycle starts again. If the check is respected, the full dynamic can be recovered.

- **Full Dynamic Computation**: The thrust profile *T(x)*, mass profile *m(x)* and time history *t(x)* are obtained numerically integrating the equations of motion.

The shape-based algorithm presented works with MEE for ease and to avoid singularities. However, such elements have weak physical meaning with respect to other possible orbit representations and therefore the orbits in this thesis will be expressed in KP for easier comprehension.

A completely analytical shape based approach was necessary due to its really low computational complexity, which makes it possible to evaluate several different trajectories in few seconds. This is of paramount importance for the scope of the multi-deployment algorithm since a really large number of transfers have to be computed as fast as possible. As already stated in the introduction to the problem, increasing the size of the problem the dimension of the solution space of the optimization critically increases. Despite the choice of a heuristic method to perform the optimizations, the needed function evaluations are still numerous and therefore the selection of a fast algorithm for the low-thrust trajectory computation was of paramount importance.

Numerical example

For the sake of completeness, an example of the algorithm output for a single transfer is here reported. The transfer was studied for three different values of thrust-to-mass ratio. A case of LEO to GEO orbit raising has been considered. The KP of the initial and final orbits are reported in Table 4.1. For all the three cases a spacecraft with initial mass $M_0 = 100\ kg$ and specific impulse $I_{sp} = 3500\ s$ was considered.

	a [DU]	e [-]	i [deg]	Ω [deg]	ω [deg]	θ [deg]
LEO	1.3136	0.1	30	0	0	0
GEO	5.6108	0	0	0	0	0

Table 4.1: LEO and GEO keplerian elements.

The results of the three runs of the shape-based algorithm with the three different thrust levels are reported in Table 4.2. The propellant consumption

T [N]	M_{prop} [kg]	TOF [d]	N_{rev} [-]
10	13.88	1.2589	5
1	13.89	11.3462	46
0.1	13.89	107.5011	452

Table 4.2: LEO-GEO transfer results.

stays the same since the energetic change necessary depends only on the departure and target orbit. Obviously, by decreasing the maximum thrust, and consequently the maximum acceleration of the spacecraft, the TOF of the transfers increases in the same way. The same happens for the number of revolutions, which also increases by about one order of magnitude together with the TOF.

Only the trajectory and control law of the case which needs fewer revolutions around the attractor is here reported to make the solution more easily interpretable. The trajectory of the transfer is shown in Figure 4.3 while in Figure 4.4 the control law recovered at the end of the shape-based algorithm is represented. T is the thrust in absolute value, while T_{in} and T_{out} are respectively the in-plane and out-of-plane components. Rev_1 indicates the first revolution around Earth, Rev_{int} each of the intermediate ones while Rev_{last} the last one. It is possible to assess how the maximum thrust constraint of 10 N is respected for each revolution.

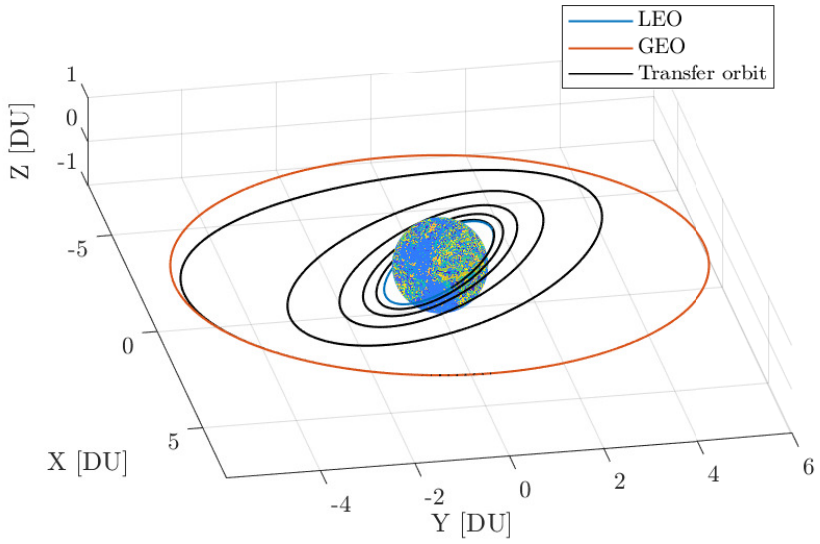


Figure 4.3: LEO-GEO transfer trajectory.

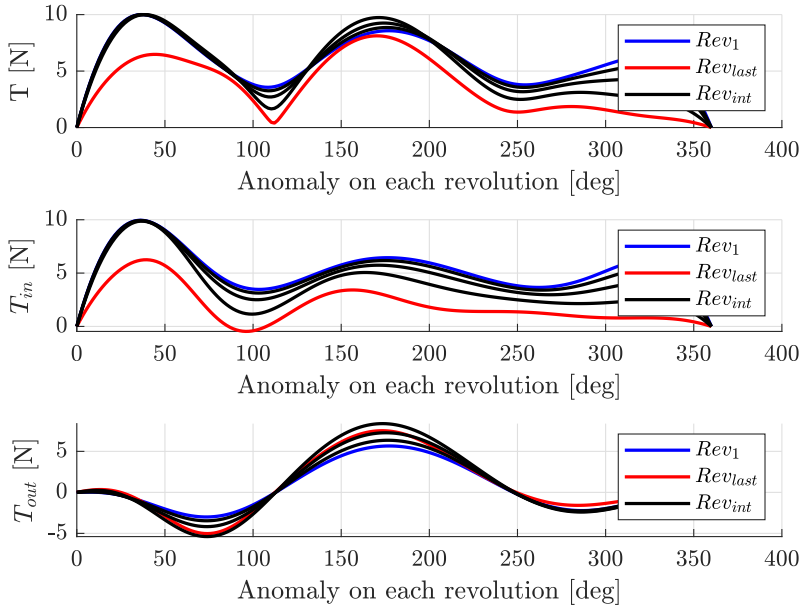


Figure 4.4: LEO-GEO transfer control law.

4.1.2 Target anomaly

The shape-based algorithm presented in section 4.1.1 matches the target state KP_2 given as input in all its components. However, the correct anomaly at which to release a satellite depends on the time of the arrival and therefore it is not necessarily equal to θ_2 (the sixth element of the input vector KP_2). The impact of the value of the target anomaly θ_2 on the cost and duration of low-thrust transfers is shown by Figure 4.5. For the analysis, for instance, an arbitrary starting orbit with keplerian elements presented in Table 4.3 was selected.

a_1 [DU]	e_1 [-]	i_1 [deg]	Ω_1 [deg]	ω_1 [deg]	θ_1 [deg]
1.1	0.1	67	60	0	0

Table 4.3: KP of the starting orbit for the analysis of the target anomaly impact on low-thrust trajectories.

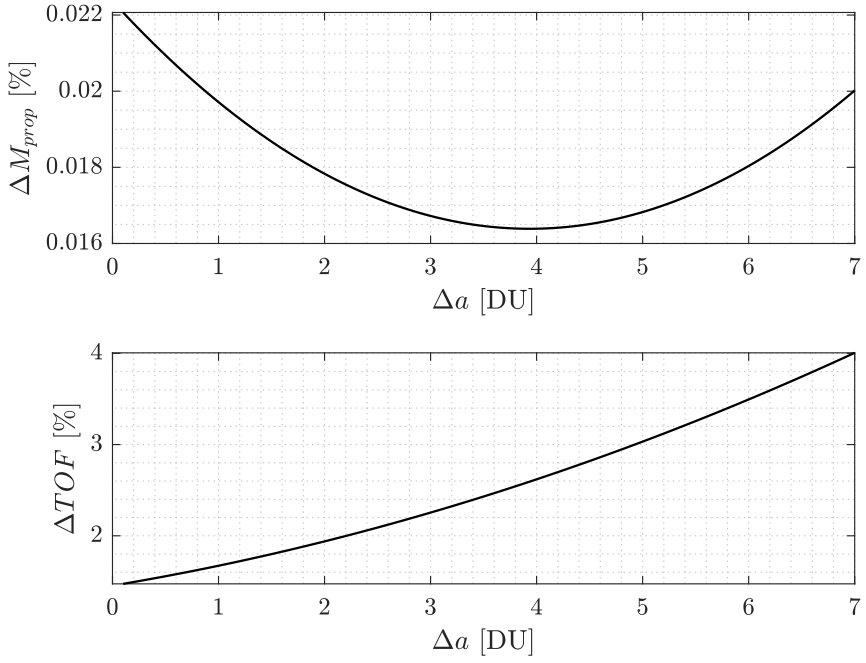


Figure 4.5: Maximum percentage change in propellant and TOF varying target anomaly and target semi-major axis.

For simplicity, the target orbit has been chosen to have the same elements as the starting one, apart from a_2 and θ_2. Different target orbits are taken

into consideration during the analysis, with semimajor-axis $a_2 = a_1 + \Delta a$. At each value of a_2, the propellant and TOF of the transfer, with θ_2 ranging from 0° to 360°, have been computed. The average of the results for each θ_2 was computed and the maximum percentage difference in absolute value was kept. This has been repeated for each value of a_2 and the interpolation of the results is presented in Figure 4.5. It is clear how the propellant cost experiences nearly no impact by the variation of the target anomaly. The TOF has a slightly greater dependence from θ_2, which stays however confined inside reasonable percentage bounds.

A low-thrust transfer in planetocentric scenarios involves many revolutions around the main attractor to reach the target. For this reason, reaching the correct phasing on an orbit is just a control problem and does not deeply affect the cost or the duration of the transfer.

In conclusion, it was confirmed that the scope of estimating the duration and propellant consumption of the mission can be fully achieved without considering the exact anomaly at which to reach the target orbit.

4.1.3 Improvements to shape based algorithm

The shape-based algorithm presented in section 4.1.1 only deals with keplerian motion. At the current state of the art, the environmental perturbations could be accounted a posteriori only in the case in which the perturbing acceleration was at least about one order of magnitude less than the acceleration provided by the thrusters. When dealing with orbits in the LEO region, environmental perturbations have an important role when dealing with low-thrust, since the order of magnitude of the perturbing accelerations are sometimes the same as (if not higher than) the thrust acceleration, as it can be seen from Figure 4.6 [38]. In the latter for each effect the logarithm of the disturbing acceleration, normalized to 1 g, is shown as a function of altitude. For this reason, the impact of the perturbations, especially the J_2 effect, on the spacecraft trajectories must be taken into account when planning the transfers, since it cannot be simply counteracted and canceled by the spacecraft.

The shape-based algorithm workflow in Figure 4.2 is then modified to take into account the J_2 perturbation disturbances. The new workflow is schematized in Figure 4.7.

The only difference compared to the previous version is the introduction of the perturbing block, whose functioning is here described:

- J_2 **Perturbation**: The impact of the Earth gravitational perturbation is twofold and both the effects are accounted for in this block.

    1. First, the J_2 perturbation obviously has an effect on the spacecraft trajectory. The most accurate way to consider this perturbation would be considering its punctual effect on each revolution around

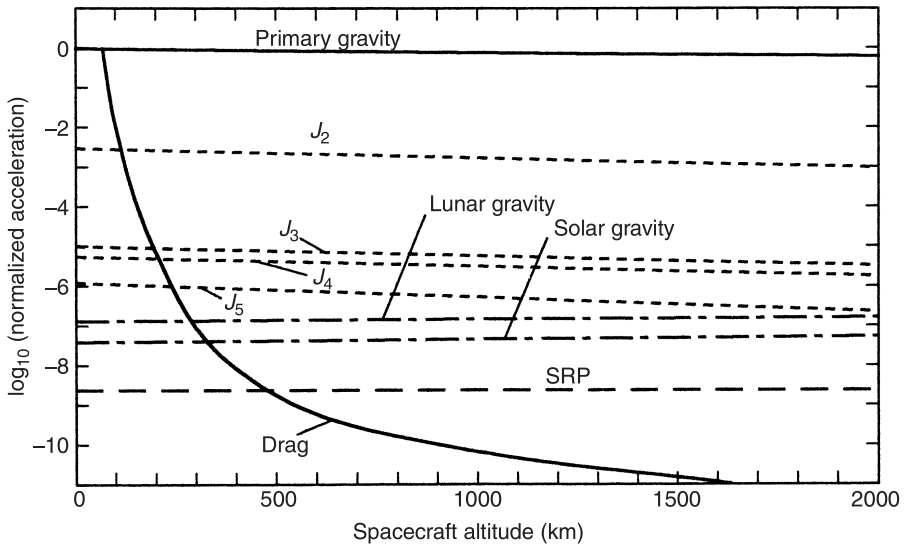


Figure 4.6: Comparisons of the disturbing accelerations for the main sources of perturbation.

the main attractor. However, this approach would need the integration of the equations of motion which would dramatically increase the computational load. In addition to this, the adoption of a shape-based algorithm, which recovers the dynamics from a pre-imposed shape, further complicates the introduction of such accelerations a posteriori. For this reason, only the integral of the effects of the J_2 perturbation are considered in the trajectory computation. Once target current target state MEE_k is reached, the RAAN of the current state is corrected with the secular effect of the perturbation over that revolution. The validity of this approximation is discussed in the next paragraphs.

2. Second, as stated in section 2.1.3, the J_2 perturbation affects the target state position. When dealing with impulsive maneuvers, the Lambert theorem allows to fix the TOF a priori and therefore the exact target state position can be computed to perform the transfer. The adopted shape-based approach does not allow to know the TOF before the trajectory computation. For this reason, after each revolution (of the many which compose a low-thrust transfer) the target state also is updated with the secular effect of the J_2 perturbation. A tolerance is introduced on the RAAN, such that when $|\Omega_k - \Omega_2|$ is less than this value, the target state is considered to be reached.

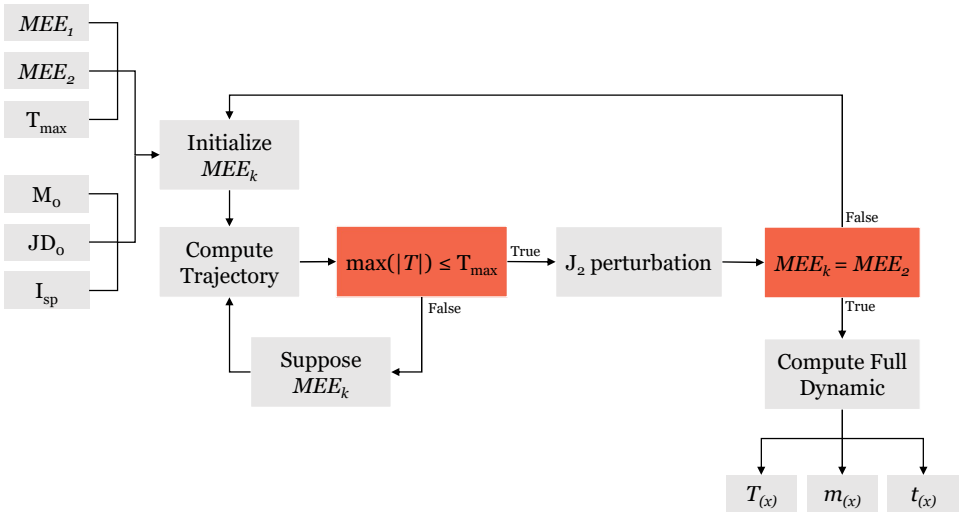


Figure 4.7: Shape-based algorithm workflow with J2 correction.

While correcting the current state with the secular effect of the perturbation introduces some discontinuities in the trajectory and control low, it provides a good estimation of the propellant consumption and transfer duration with a really low computational load, which is the main driver of the algorithm development. To assess the validity of such an approach, the results were compared to an integration of the equations of motion of the spacecraft, considering both the J2 punctual accelerations and the control low found from the shape-based approach. A generic starting orbit was selected for this analysis, whose parameters are reported in Table 4.4, while the semi-major axis was left to vary between 1.1 and 2 DU. The target orbit of the single revolution was considered to be an orbit with the same parameters but with a semi-major axis 0.5% greater.

e [-]	i [deg]	Ω [deg]	ω [deg]	θ [deg]
0.1	70	78	5	0

Table 4.4: KP of the starting orbit for comparison between integration and shape-based approach with and without J2 correction.

The results of the comparison are presented in Figure 4.8. Special consideration has been given to the RAAN error, which is one of the most affected parameters by the J2 effect and also the most expensive to eventually compensate by thrusting the spacecraft. The error in the RAAN $\Delta\Omega$ is shown in absolute value on the y-axis and represents the difference between the shape-based algorithm final state RAAN and the integration final

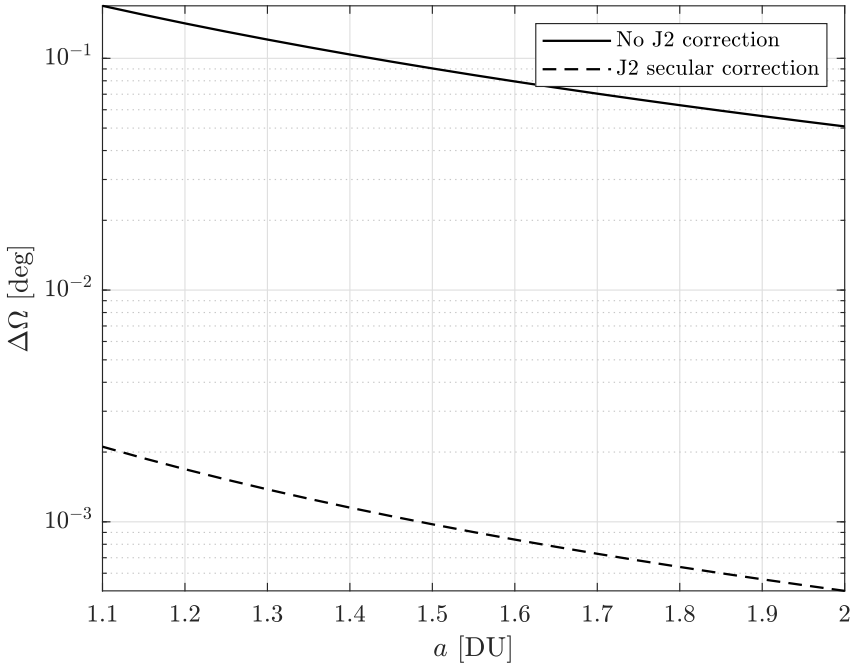


Figure 4.8: Shape-based approach error in final state RAAN with and without J2 correction compared to the integration of the control law.

state RAAN, integration performed considering the punctual J_2 accelerations. The solid line in Figure 4.8 shows how the error in the target RAAN would be two orders of magnitude higher without the correction and would accumulate over the many revolutions of the transfer. The J2 correction allows to take into account the secular effect of the perturbation and therefore keeps the error on the final state limited to the short-period and long-period variations of the parameter.

4.2 J_2 exploiting transfer strategy

The same transfer strategy to perform the transfers with consistent changes in RAAN presented in section 3.2 is adopted also in the low-thrust scenario. The workflow of the transfer strategy is equal to the impulsive one, apart from the transfers of course, and is reported again in Figure 4.9. However, some differences inside some of the blocks must be highlighted.

- **Inputs**: The starting orbit KP_1 and the target orbit KP_2 are the two main inputs to the function. This time there is no TOF anymore since

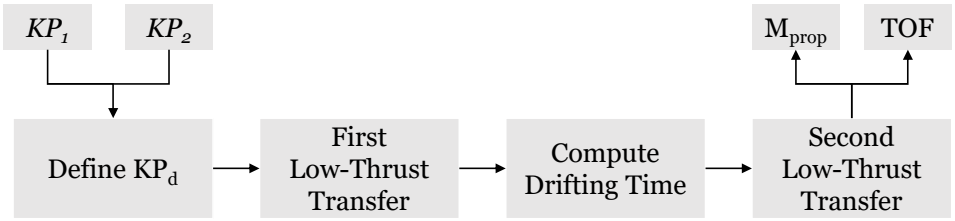


Figure 4.9: Single low-thrust J2-exploiting transfer scheme.

it is not a variable of the optimization but an output of the shape-based algorithm.

- **Drifting Orbit Definition**: The drifting orbit in the low-thrust transfer is defined in a different way. The 6 KP are reported in Table 4.5.

a_d	e_d	i_d	Ω_d	ω_d	θ_d
var	0	*var*	*free*	$\frac{1}{2}(\omega_1 + \omega_2)$	*free*

Table 4.5: KP of the drifting orbit for low-thrust J2-exploiting transfer strategy.

The two differences compared to Table 3.1 are in the RAAN and the anomaly:

  - Saying that Ω_d is left *free* means that no control is considered on the final value of the RAAN when moving to the drifting orbit. The RAAN of the spacecraft is left to vary according to the J2 effect. There would be no point or convenience in controlling the RAAN since the goal itself of reaching the drifting orbit is to change such parameter by exploiting the natural perturbations instead of the propellant.
  - θ_d is *left* free since it affects neither the secular RAAN variation nor the cost or duration of the transfer, as proved before.

- **First Transfer**: Once the drifting orbit is defined, the cost and duration of the first low-thrust transfer are computed.

- **Drifting Time Computation**: Once the target orbit has been reached, it is necessary to compute the amount of time necessary to close the RAAN gap. While what said for the impulsive transfer strategy was true for that kind of maneuvers, when dealing with low-thrust transfers the definition of the exact Ω_d at which to depart from the drifting

orbit is slightly more complicated. Since low-thrust transfers between the orbits have large times of flight, an amount of relative RAAN shift happens also during the transfer from the drifting to the target orbit. For this reason, departing when the two values of the RAAN are already equal would require some degree of control on such parameter during the transfer to the target orbit to keep such values equal. The most efficient and less expensive way to perform the transfer would be to estimate the relative drift which takes place during the transfer from the drifting to the target orbit and to take it into account when computing the drifting time. To adopt this approach the following steps must be taken:

- A fictitious target orbit is defined with the same parameters as the true target orbit apart from the RAAN, which is left free, in order to estimate the amount of shift that would take place during the transfer. Such shift is referred to as $\Delta\Omega_{t,1}$ and is computed by evaluating at the end of the trajectory computation how much the J2 effect made the RAAN change.
- Once computed the transfer to the fictitious target orbit, knowing the TOF of the transfer it is possible to estimate also how much the target orbit shifts during the duration of the transfer. This amount of RAAN shift is defined as $\Delta\Omega_{t,2}$.
- From these two quantities it is possible to compute the amount of relative shift $\Delta\Omega_t$ (eq. (4.1a)) and use it to correct the RAAN gap to close (eq. (4.1b)). The time necessary is then computed according to eq. (3.2c) and with eq. (2.3) the new values of the RAAN at the end of the waiting time are updated.

$$\Delta\Omega_t = \Delta\Omega_{t,1} - \Delta\Omega_{t,2} \tag{4.1a}$$

$$\Delta\Omega = \Omega_2 - \Omega_d + \Delta\Omega_t \tag{4.1b}$$

Shortly, this correction on $\Delta\Omega$ allows finding the correct Ω_d at which to depart from the drifting orbit, which does not coincide with Ω_2 unless the two orbits are so close or so high in altitude that the J2 effect during the transfer can be considered negligible. Thanks to this estimation, no propellant has to be consumed to control the RAAN because the transfer is planned in such a way to meet the target orbit at the correct value of this parameter.

- **Second Transfer**: Once the RAAN of current and target orbits have been updated after the stationing onto the drifting orbit, the second low-thrust transfer to the target orbit can effectively be computed.

- **Outputs**: At the end, the whole cost of the two-legs transfer is computed. In terms of propellant, the total amount needed to reach the target orbit is the sum of the propellant consumed in the two legs of the transfer. In terms of time, the total duration of the transfer is the sum of the TOFs of the two legs and also the waiting time onto the drifting orbit.

Numerical results

As a first example of this low-thrust transfer strategy, the same problem investigated in section 3.2 for the impulsive strategy was taken into consideration. The parameters of the two orbits are the ones in Table 3.2 and the spacecraft has again initial mass $M_0 = 100\ kg$, but this time with a specific impulse I_{sp} of 3500 s and a maximum available thrust of 0.1 N.

For the optimization, $N_p = 10$ and $M_{gen} = 20$ were chosen. These values are lower comparing to the one used in the optimization of the impulsive transfer since this time TOF is not a variable of the optimization. Even if the trajectory computation through the shape-based is slower than the Lambert transfer, the TOF as a variable strongly affects the final result and therefore more generations are necessary in the impulsive transfer to find good solutions. The variables of the optimization, as already stated in Table 4.5, were the semi-major axis and the inclination of the drifting orbit. The lower and upper bounds, lb and ub, are the same as in Table 3.3, reported again in Table 4.6.

Parameter	lb	ub
a_d [DU]	1.05	1.5
i_d [deg]	67	67

Table 4.6: Lower and upper bounds for the low-thrust transfer strategy example.

The final results of the optimization are presented in Figure 4.10, where all the non-dominated solutions found by the optimization method are stored. The Pareto front does not present its typical shape due to the strong non-linearity of the problem. For instance, the Pareto front would present itself as the one in Figure 2.2 if adopting the same transfer strategy but to move between orbits which are fixed in space, which means that do not experience the RAAN variation explained in section 2.1.3.

It is interesting to focus on one of the solutions, such as the fastest one, which is the one in the lower right of the Pareto front. The results in detail are shown in Table 4.7. The drifting orbit in this case has a semi-major axis a_d equal to 1.1432 DU while the inclination is fixed at 67°.

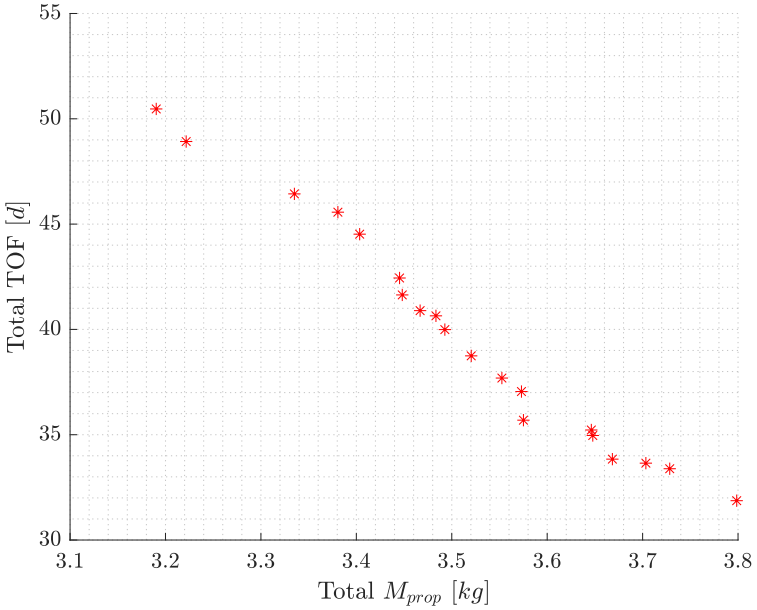


Figure 4.10: J2-exploiting low-thrust transfer strategy example results ($M_0 = 100\ kg$; $I_{sp} = 3500\ s$).

	M_{prop} $[kg]$	TOF $[d]$
Transfer 1	2.51	24.80
Drifting	-	0.24
Transfer 2	1.28	6.82
Total	3.79	31.86

Table 4.7: Fastest solution details of the low-thrust transfer strategy example.

When using low-thrust propulsion, a relevant portion of the total TOF of the missions is made by the two transfers. In some peculiar cases, the transfers may also take the greater part of the TOF, as it can be seen in Table 4.7. When dealing with impulsive maneuvers, the times of flight of the transfers are much shorter and therefore the greatest part of the duration of the mission is the drifting time. However, as it can be seen from Figure 3.3, this does not necessarily lead to improvements in the duration of the mission due to two main reasons. First, no portion of the RAAN gap may be covered during the transfers due to their short duration. In addition to this, the fact that the transfers are more expensive makes it less convenient to go far from

the target orbit in order to make the RAAN shift happen faster.

The Pareto fronts of Figure 3.3 and Figure 4.10 are compared in Figure 4.11. The advantages in adopting low-thrust propulsion are flagrant,

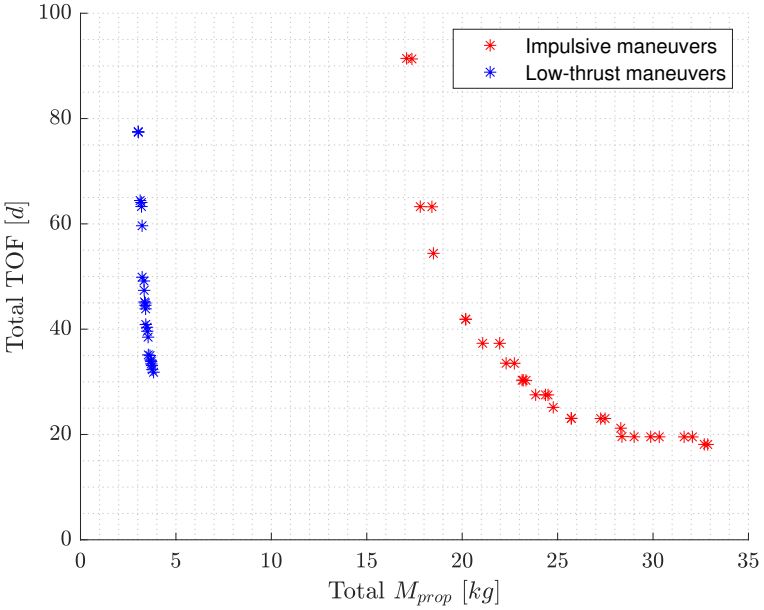


Figure 4.11: J2-exploiting transfer strategy example results comparison ($M_0 = 100\ kg$).

since the low-thrust Pareto achieves the same TOF with way lower propellant necessary. While a slight reduction in the minimum total TOF of the mission may be achieved, the propellant cost of one single transfer in an impulsive maneuvers scenario becomes critically high.

It is interesting to repeat the same example as before, again in low-thrust but this time with $\Omega_1 = 0°$ and $\Omega_2 = 30°$. Since the starting orbit is below the target one, $\Delta\dot{\Omega} = \dot{\Omega}_1 - \dot{\Omega}_2$ will be a negative quantity. In the previous case this rate of precession was coherent with the amount of shift needed since it was $\Omega_1 - \Omega_2 = -30°$. In this second example, the direction of the shift needed is opposite, meaning that with the latter shift rate the RAAN gap to cover would be 350° instead of 30°. One faster option, but more expensive, would be to choose a drifting orbit with a larger semi-major axis than the target one, in order to have a $\Delta\dot{\Omega} > 0$.

The results of the optimization of this second example, performed with the same tuning parameters and bounds, are shown in Figure 4.12. It is clear how the total time of flight of the mission increases in this case. All the solutions presented in the figure adopted the second strategy mentioned before, which means selecting $a_d > a_2$. In particular, the fastest solution

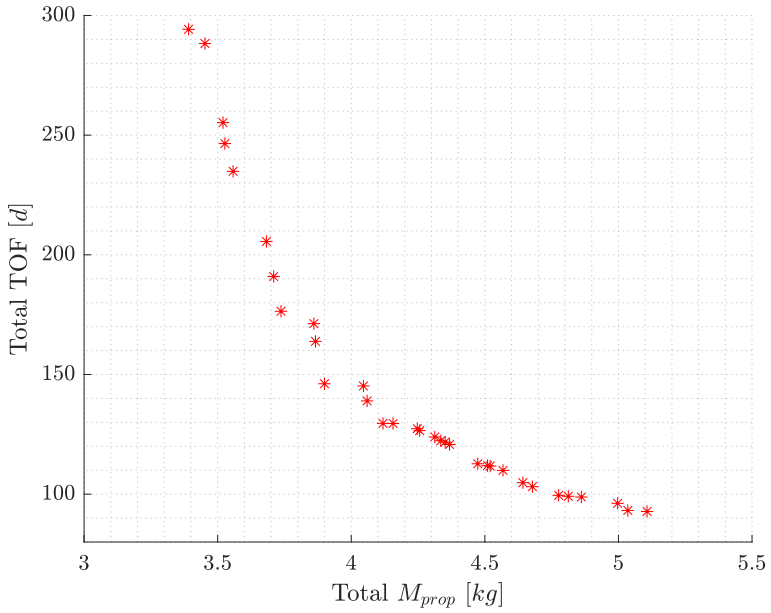


Figure 4.12: Low-thrust transfer strategy variant example results ($M_0 = 100$ kg; $I_{sp} = 3500$ s).

presented $a_d = 1.5$ DU, suggesting that increasing the value of the upper bound of the solution space would lead to faster solutions at the cost of an increase in propellant consumption. The slowest solution in figure presents $a_d = 1.34$. The solutions characterized by $a_d < a_2$ resulted in much higher TOFs and were not reported in the figure. These two examples highlight the importance of the planning of the multi-deployment strategy and in particular of the releasing order of the satellites when dealing with different RAAN values.

4.3 Multi-insertion on same orbit

Conceptually, phasing maneuvers in low-thrust work the same way as in section 3.3. However, the decreasing/increasing of the semi-major axis, in this case, is not instantaneous and therefore part of the anomaly gap is already covered during the transfer from the original to the intermediate orbit. Low-thrust phasing maneuvers could be object of the optimization as well as the single transfers between orbits. The semi-major axis of the intermediate orbit indeed cannot be defined analytically with an equation like eq. (3.3), but depends on the duration of the single transfers (which is not known a priori using the free-time-of-flight shape-based algorithm). One way to solve the problem may be performing a optimization with a

semi-major axis of the intermediate orbit and waiting time on the latter as variables of the optimization to perfectly rendezvous the target on the way back to the original orbit. Even though this would be the most accurate way to approach the problem, this is not the most suited way to implement the phasing maneuver inside the multi-deployment algorithm. Such a way would indeed introduce two additional variables for each phasing to perform, eventually critically increasing the size of the optimization problem. Since the prime goal of the algorithm is to define a multi-deployment strategy and assess its cost and duration, at this stage the phasing problem can be treated in a less detailed way and then be optimized a posteriori. The orders of magnitude of cost and duration of the phasing maneuvers are indeed smaller than the ones of the overall multi-deployment mission and therefore their approximation does not affect the validity of the results of the multi-deployment algorithm.

For this reason, a fast and efficient way to approach and approximate the phasing problem was selected. The possibility to compute the propellant mass needed for the low-thrust phasing starting from the Δv of the impulsive maneuver was investigated. The difference between the cost of the Hohmann maneuver, Δv_H, and the one needed to perform the same maneuver in low-thrust, Δv_{LT}, is represented by the parameter GL, shown in eq. (4.2).

$$GL = 100 \cdot \left(\frac{\Delta v_{LT}}{\Delta v_H} - 1 \right) \tag{4.2}$$

The gravity loss GL is an important parameter of merit of a low-thrust trajectory: it shows how a specific low-thrust maneuver is more expensive in terms of Δv when compared to the corresponding impulsive one. It is necessary to study the behaviour of this parameter. A parametric analysis was performed, whose results are shown in Figure 4.13. The analysis was carried on assuming a change in the semi-major axis of 10% of its nominal value but changing this percentage does not affect the trend shown in the figure. The behaviour of GL was studied for different values of a and e. It can be seen how the loss is under 10% for values of eccentricity less than 0.3 and almost null for nearly-circular orbits.

Numerical example

Such methodology was compared to the results of a problem of design and optimization of low-thrust orbital phasing maneuver already fully solved in literature [39], where a preliminary design method is developed to estimate the cost and duration of a low-thrust phasing maneuver. As an example, a GEO phasing mission was taken into account. The KP of the GEO are presented in Table 4.8. Since the value of a_1 is large and the duration of a revolution on the orbit is long, a lower bound for a_2 not so close to a_1 has to be chosen to keep the number of revolutions to wait relatively low. It was

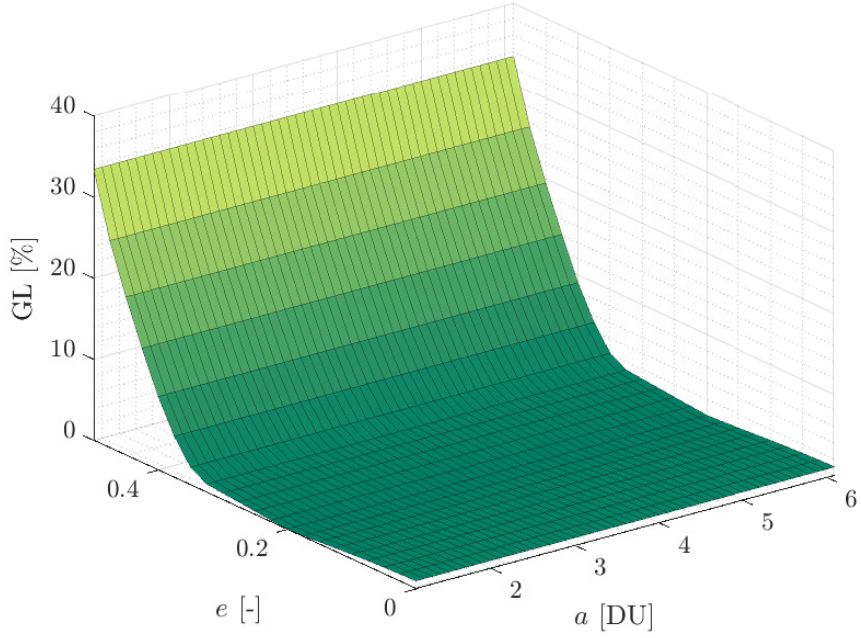


Figure 4.13: Gravity losses for semi-major axis change maneuvers.

a [DU]	e [-]	i [deg]	Ω [deg]	ω [deg]
6.6109	0	0	0	0

Table 4.8: KP of GEO for phasing maneuver example.

chosen therefore to select a number of revolutions N_{rev} such that $a_2 > 0.9 \cdot a_1$. The initial mass of the chaser is 1000 kg, and the specific impulse of the electric thruster is 3000 s. The results of the preliminary design method are compared to the ones of the impulsive approximation presented in this thesis in Table 4.9, where the phase angle indicates the amount of phasing necessary. In all of the three cases the chaser is considered to be behind the target.

With respect to the method developed by Shang et al. [39], the present impulsive approximation underestimates the duration of the phasing while overestimating the propellant mass. The underestimation of the time of flight does not impact the multi-deployment results since the order of magnitude is much lower than the one of the whole duration of the mission. This impulsive approximation is a preliminary way to take into account phasing maneuvers into the multi-deployment algorithm and therefore it is important that the

Phase angle [deg]	30	90	150
Phasing time by impulsive approximation [d]	1.8696	2.6795	3.5034
Phasing time by preliminary design method [d]	3.5985	6.2000	7.9760
Propellant consumption by impulsive approximation [kg]	6.1383	9.4832	10.6436
Propellant consumption by preliminary design method [kg]	3.0720	5.3850	6.9560

Table 4.9: GEO phasing mission results comparison.

results found are in the same order of magnitude as the ones found through an optimization of the maneuver. Once the best multi-deployment strategy is defined, the various phasing maneuvers can be treated more in detail and optimized.

4.4 Numerical results

Some examples of the functioning of the low-thrust multi-deployment algorithm and the highlight of its main features are here reported. Also, an example introducing the constraints in the way presented in section 2.2.3 is reported.

4.4.1 Multi-deployment mission optimization

A generic set of 6 satellites to deploy in LEO was considered. The KP of the orbits into which to insert each of them are reported in Table 4.10, together

ID	a [DU]	e [-]	i [deg]	Ω [deg]	ω [deg]
0	1.05	0.02	66	0	0
1	1.06	0.01	67	10	5
2	1.07	0.02	66	8	0
3	1.08	0.01	67	328	3
4	1.09	0.03	66	161	0
5	1.10	0	68	22	0
6	1.11	0.05	66	159	20

Table 4.10: Initial states for multi-deployment mission example.

with the starting orbit, indicated by the index 0, from which the vehicle starts the journey. All the KP reported are the ones at the time of the departure from orbit KP_0. The actual state of insertion will be different according to the J2 perturbation.

Since the size of the problem increases with respect to the optimization of a single transfer, the population size N_p and the number of maximum generations M_{gen} must increase as well with respect to the ones in Table 4.6 to guarantee good results. $N_p = 20$ and $M_{gen} = 30$ were chosen. The lower and upper bounds of the optimization for this example are the ones in Table 4.11. The inclinations of the drifting orbits are allowed to vary between the minimum and maximum value of the inclinations of the sets of orbits into which to release the satellites.

Parameter	lb	ub
a_d [DU]	1.0314	1.300
i_d [deg]	66	68

Table 4.11: Optimization tuning parameters for multi-deployment mission example.

The characteristics of the deploying vehicle chosen for this example are presented in Table 4.12. All the 6 satellites to be released have been considered to be nanosatellites with 5 kg mass each.

M_0 [kg]	T [N]	I_{sp}
100	0.5	3500

Table 4.12: Spacecraft characteristics for multi-deployment mission example.

The results of the optimization are visible in Figure 4.14. The two different colors represent solutions with different releasing orders, identified by different vectors p. All the solutions found from this run of the optimization algorithm have releasing order $p = [1, 2, 3, 4, 5, 6]$ apart from one, which instead releases the satellites in the order $p = [1, 2, 3, 4, 6, 5]$.

It is interesting also in this case to focus on one solution. Again, the fastest solution found from the algorithm will be analysed in detail, whose final objectives are reported in Table 4.13. The details of the single releases (indicated by the capital letter R) are reported in Table 4.14. For instance, R1 indicates the first release, which corresponds to the first element of the vector p. For each transfer, the pedices 1 and 2 respectively indicates the first and second leg of the transfer.

It is clear that also in the fastest route possible found by the algorithm,

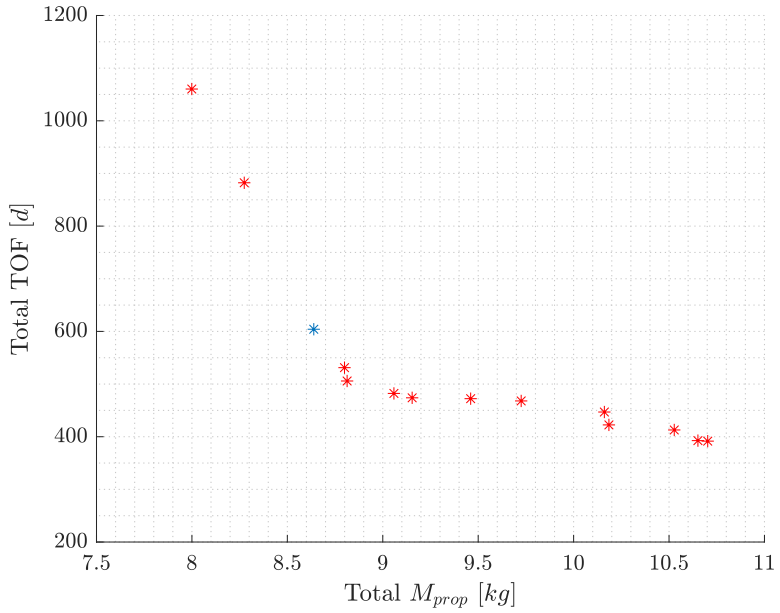


Figure 4.14: Multi-deployment mission example results ($M_0 = 100\ kg$; $I_{sp} = 3500\ s$).

M_{prop} [kg]	TOF [d]
10.70	391.6480

Table 4.13: Final objectives of the fastest solution for multi-deployment mission example.

	R1	R2	R3	R4	R5	R6
a_d [DU]	1.0948	1.0326	1.0375	1.3000	1.0314	1.2698
i_d [deg]	66.44	66.60	66.54	66.71	66.37	67.63
$M_{prop,1}$ [kg]	0.75	0.43	0.62	1.58	0.69	1.01
TOF_1 [d]	3.2585	1.8196	2.6659	4.4483	3.4542	2.8178
t_{wait} [d]	34.8707	6.5801	52.1309	127.2152	59.5546	70.5217
$M_{prop,2}$ [kg]	0.55	0.69	0.51	1.59	0.91	1.37
TOF_2 [d]	2.2958	3.1638	1.9813	5.6794	3.5002	5.6899

Table 4.14: Release details of the fastest solution for multi-deployment mission example.

the vast majority of the mission time is made up by the time needed to shift the RAAN. In particular, the most time-requiring transfer is the fourth one, which also happens to be the one to release the satellite of index 4 after having released the satellite 3, according to the releasing order. It can be seen from Table 4.10 that the orbits 3 and 4 have a large initial difference in RAAN, which justifies the long wait necessary to nearly (section 4.2) match the two values of such parameter. Also, the semi-major axis of the drifting orbit for this transfer was chosen to be $a_d = 1.3$, which is also the value of the upper bound given to this variable in the optimization. This choice suggests that a higher semi-major axis may be chosen to speed up the drift at the cost of higher consumption as well. This transfer is already the most expensive one of the six since it required a consistent orbit raising to reach the desired drifting orbit.

4.4.2 Multi-deployment mission constrained optimization

It is interesting to study the problem in the case in which some constraints are applied. One of the mission constraints that may be necessary is the maximum releasing time of one or more of the satellites. Also, there may be a maximum mission duration, that is a limit date at which the last satellite has to be released. The same example as before was studied again, but this time with the constraints in Table 4.15.

Constraint	Value $[d]$
Max mission duration	730
Max release time ID 5	90

Table 4.15: Constraints for multi-deployment mission constrained example.

The result of the constrained optimization is the Pareto front in Figure 4.15. As expected, the results are worst with respect to the unconstrained problem since the solutions respecting the constraints are a subset of all the solutions. Differently from the releasing orders of the unconstrained solutions, which released the satellite on the orbit 5 as last or second last, all the solutions in Figure 4.15 release it as first to respect the constraint in Table 4.15. The three releasing orders of the solutions are reported in Table 4.16.

4.4.3 Heuristic - hybrid optimization approaches comparison

Even if the approach in section 2.2.2 was developed to allow the solution of cases with $N > 11$, the adoption of this approach for cases with $N < 11$ was also investigated. This approach reduces the search space in each step, allowing the heuristic algorithm to focus only on the best solutions. For

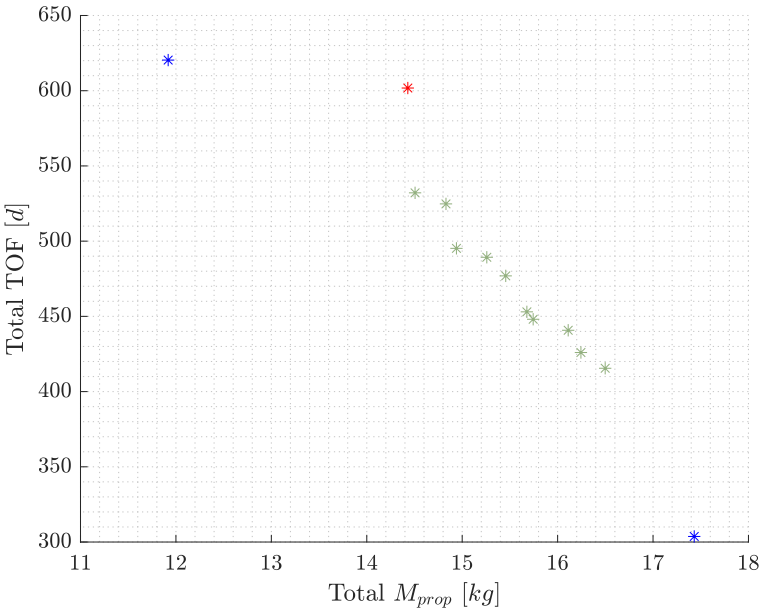


Figure 4.15: Multi-deployment mission constrained example results ($M_0 = 100\ kg$; $I_{sp} = 3500\ s$).

Color	Releasing order
Red	[5,4,3,2,6,1]
Blue	[5,4,6,2,1,3]
Green	[5,6,2,1,3,4]

Table 4.16: Multi-deployment mission constrained example releasing orders.

this reason, it was interesting to compare the two approaches in terms of computational time and quality of the solutions. Since the final result of a heuristic optimization method strongly depends on the initial solution, comparing only one optimization run to compare the results would not be really significant. For this reason, in order to truly assess whether or not one approach is better than the other, more than one run per each must be considered. For both of them, 10 runs were performed and the 10 Pareto fronts have been merged into one, where only the non-dominated solutions survived.

The same example which was analyzed in section 4.4.1 is used here for the comparison. The hybrid approach was carried on with $r = 3$, meaning that two iterations were needed to find the final solutions. The 10 runs of the pure heuristic method were performed with $N_p = 20$ and $M_{gen} = 20$. About the

hybrid approach, two different values of these two parameters were chosen for the first and second iteration. This was done since the second iteration has a solutions space smaller than the first iteration. While the first iteration has 120 possible permutations according to eq. (2.6), the second iteration only has 6, since the matrix **P** in this case only has N! possible columns (eq. (2.1)), with this time $N = r = 3$. The two sets of values are reported in Table 4.17. The upper and lower bounds are the same as in Table 4.11.

	Iter 1	Iter 2
N_p	10	5
M_{gen}	20	10

Table 4.17: Optimization tuning parameters of hybrid approach for optimization approaches comparison.

The two Pareto fronts are shown in Figure 4.16, while the transparent

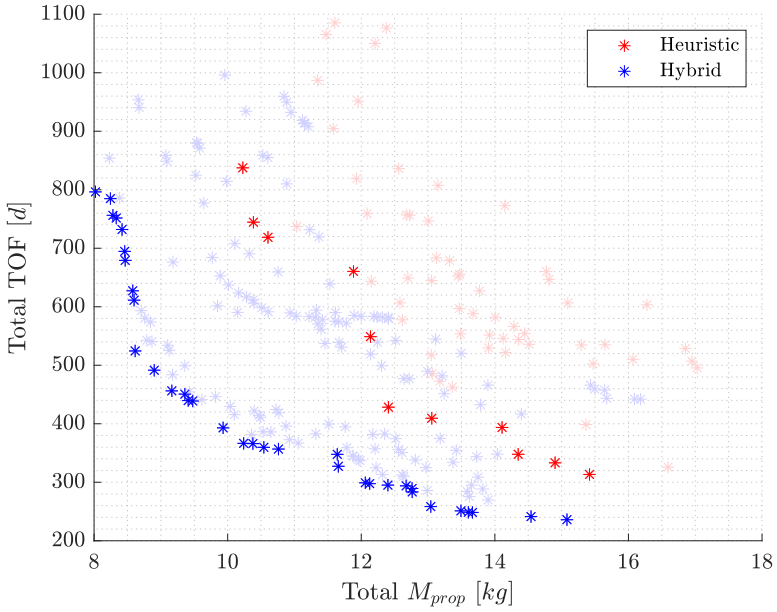


Figure 4.16: Comparison of optimization approaches results ($M_0 = 100\ kg$; $I_{sp} = 3500\ s$).

markers represent the clouds of solutions found by the 10 runs of the optimization for both the approaches. It is clear how the Pareto front found from the hybrid optimization approach is composed by far better solutions with respect to the pure heuristic one. Indeed, the blue Pareto front dominates the red one in all its solutions found.

It is important when comparing the two optimization approaches to compare not only the final results but also the computational effort necessary to achieve them. The parameters which tune the population size N_p and the maximum generation M_{gen} were chosen to have a similar duration of the runs between the two approaches. The time is indeed proportional to the function evaluations necessary to find the solution. The average duration of the 10 runs of the hybrid approach was 225.5 s, while the heuristic one needed an average of 282.3 s per run.

It is possible to conclude that the hybrid approach outperforms the pure heuristic one both in terms of results and computing effort. In light of such results, the hybrid optimization approach will be chosen as the standard to perform the minimization of the objectives, also when dealing with problems characterized by $N < 11$.

5. Simulation results

Due to the flagrant advantages in performing the transfers in the J_2 exploiting transfer strategy with low-thrust propulsion, only this propulsive class will be considered for the simulations of this chapter. The latter is organized as below:

- First, the most important features considered for the releasing vehicle are presented.
- Second, applications of the algorithm to a constellation deployment case are considered.
- Finally, the algorithm is applied to some multi-satellites deployment scenarios.

5.1 Vehicle definition

Before analyzing some case studies, it is necessary to exactly define the main characteristics of the multi-deployment vehicle:

- **Mass**: the dry mass of the vehicle M_v was estimated through an analytical statistical relationship [40]. In particular, a linear regression of the mass of launcher adapters in function of the supported mass M_s was considered to estimate it. The reasons behind this choice are twofold:

  1. First, the adapter is dimensioned to sustain the weight of all the supported mass on top of it, which is the worst-case scenario in terms of loads. The actual disposition of the satellites on the releasing vehicle is likely to be different and less critical in terms of loads.
  2. In addition to this, the adapter mass is dimensioned to sustain the launcher loads, which are the most critical ones.

  For these two reasons, this approximation was found to be useful for a preliminary mass allocation to the multi-deployment vehicle dry mass.

The analytic relationship is the one in eq. (5.1). The mass M_v already includes the mass of the supported mass.

$$M_v = 0.0755 \cdot M_s + 50.252 + M_s \tag{5.1}$$

- **Thruster**: the thrust level of electric thrusters nowadays ranges from μN to few tenths of N. The thrust class greater than 0.01 N is commonly used for missions of orbit raising or characterized with consistent orbit changes [41]. For this analysis, the characteristics of the RIT 2X Series [42] thruster will be taken into account. The nominal thrust of 171 mN will be considered the maximum thrust available for every single thruster and for the specific impulse the constant value of 3500 s will be considered. This class of thrusters also guarantees one of the longest lifetimes on the market with more than 2 years of proved lifetime and a total impulse higher than 10 MNs.

5.2 Constellation insertion

In this section, the cases of constellation deployment are dealt with. First, an example with an existing constellation is reported. Due to the relatively high mass of the satellites belonging to existing constellations around Earth, the deployment of only one portion of the constellation is considered. Afterward, the deployment of a whole nano-satellites constellation is studied.

5.2.1 Starlink replacement mission

When planning and then deploying a constellation around Earth, usually spare satellites are accounted for. Spare satellites are inactive satellites exactly equal to the active ones of the constellations. Usually, the spare satellites are usually held on storage orbits and kept ready to move to the working orbit and substitute one of the active ones in case of failure. However, it could happen that spare satellites are not available or that satellites with new features have to be launched and released on different orbital planes of the constellation. In this case, more than one launch would be necessary to release each satellite on the correct orbital plane. Thanks to the multi-deployment strategy analyzed in this thesis, this goal could be achieved with a single launch.

The deployment of a portion of the satellites belonging to the Starlink [43] constellation has been considered. The satellites of this constellation have a mass of 260 kg each. The Starlink spacecraft constellation will be spread into 24 orbital planes with an inclination of 53°, on circular orbits with an altitude of 550 km. A possible case may be the replacement of 6 satellites of the constellation, each on a different orbital plane. Supposing that the multi-deployment vehicle is already released on one of the orbital

ID	a [DU]	e [-]	i [deg]	Ω [deg]
0	1.0862	0	53	0
1	1.0862	0	53	15
2	1.0862	0	53	30
3	1.0862	0	53	45
4	1.0862	0	53	60
5	1.0862	0	53	90
6	1.0862	0	53	105

Table 5.1: Initial states for Starlink replacement mission.

planes of the constellation (the one referred to with ID 0), the planes onto which to release the satellites are reported in Table 5.1.

According to eq. (5.1), the dry mass of the vehicle would be $M_v = 1.728$ kg. The wet mass, which is the initial mass M_0 of the spacecraft, was rounded up to 2000 kg. The propulsive characteristics of the vehicle are the ones explained in section 5.1. Due to the high initial value of M_0, the vehicle was considered to be provided with 10 RIT 2X thrusters, for a total maximum thrust available of 1.71 N.

The optimization of the multi-deployment mission was carried on through the use of the branch and bound based heuristic approach. With the size of the problem $N = 6$ and the size of the subsearch $r = 3$, two iterations were needed to find the final solutions. The tuning parameters of the optimization are reported in Table 5.2, while the lower and upper bounds of the research are shown in Table 5.3.

	Iter 1	Iter 2
N_p	15	10
M_{gen}	30	20

Table 5.2: Optimization tuning parameters for Starlink replacement mission.

Parameter	lb	ub
a_d [DU]	1.0314	1.250
i_d [deg]	53	53

Table 5.3: Lower and upper bounds for Starlink replacement mission.

The results of one run of the algorithm are presented in the top Pareto front in Figure 5.1. A constraint on a maximum mission time of 2 years was

imposed, reason why all the results do not exceed 700 d of duration. The bottom subfigure in Figure 5.1 represents a focus on the fastest solutions of the complete Pareto front of the top one. Also, it is possible to see that also the most expensive solution does not exceed the propellant mass that was allocated a priori to the vehicle when assuming the initial mass M_0.

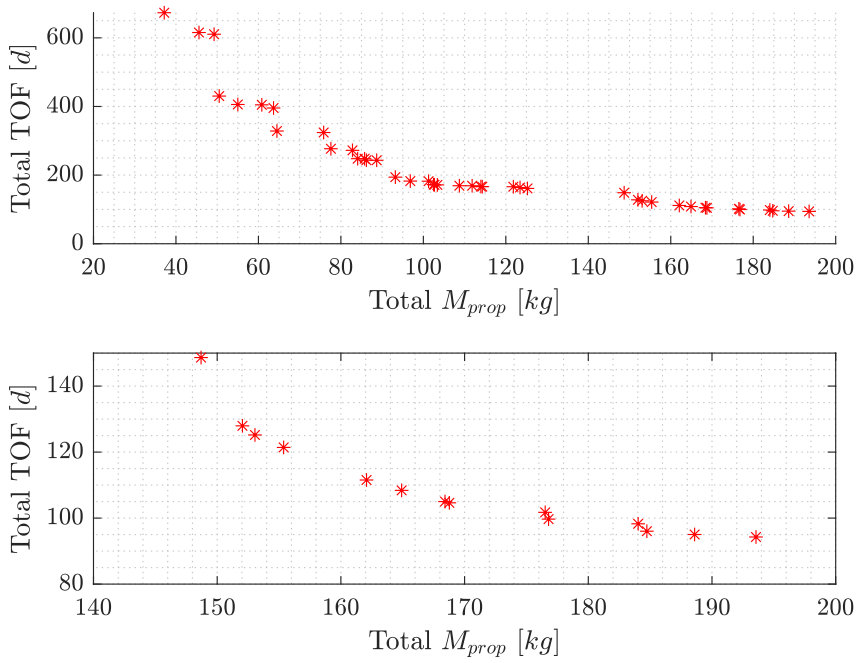


Figure 5.1: **Top**: Pareto front for Starlink replacement mission. **Bottom**: focus on fastest solutions ($M_0 = 2000\ kg$; $I_{sp} = 3500\ s$).

Let's focus again on the fastest solution. As it could be expected from Table 5.1, the fastest solutions are characterized by the releasing order $p = [1, 2, 3, 4, 5, 6]$. The release details of this solution are reported in Table 5.4. Some considerations arise by analyzing the results:

- In four out of the six transfers, the value of a_d was set equal to the upper bound of the solutions space. Again, this suggests that faster solutions, if desired, can be achieved by setting a higher upper bound.
- It can be noticed how the propellant mass required for the last transfers is smaller in magnitude than the one of the initial transfers. This is due to the lightening of the vehicle, whose main contribution is given by the release of the satellites rather than by the propellant consumption. Together with the propellant mass, also the times of

	R1	R2	R3	R4	R5	R6
a_d [DU]	1.2136	1.2500	1.1898	1.2500	1.2500	1.2500
i_d $[deg]$	53.00	53.00	53.00	53.00	53.00	53.00
$M_{prop,1}$ $[kg]$	23.44	24.91	13.40	16.14	11.86	7.68
TOF_1 $[d]$	7.1962	7.7478	4.2896	5.0999	3.7778	2.5871
t_{wait} $[d]$	2.8250	0.3743	7.7710	3.1846	13.1871	5.8489
$M_{prop,2}$ $[kg]$	23.16	24.54	13.27	15.92	11.67	7.57
TOF_2 $[d]$	7.1659	7.6668	4.2363	5.0745	3.7877	2.4532

Table 5.4: Release details of the fastest solution for Starlink replacement mission.

flight of the transfers decrease due to the higher acceleration peaks that can be achieved, being the maximum available thrust constant.

- Finally, it is possible to notice how the waiting times t_{wait} on the drifting orbits grow bigger with the going on of the mission. Due to the smaller times of flight, a smaller portion of the RAAN gap is covered during the two legs of the transfers. For this reason, more time must be spent on the drifting orbit to obtain the desired RAAN shift.

Due to the really high initial mass of the vehicle, an extreme case with 10 thrusters simultaneously firing was considered to get the previous results. It is interesting to study the behaviour of the solutions when changing the number of thrusters the vehicle is provided with. A run of the optimization with the same tuning parameters as before has been performed with 7, 5 and 2 thrusters. The resulting Pareto fronts are plotted in the same figure in Figure 5.2, where the numbers in the legend represent the number of thrusters that can be fired simultaneously. Respectively, the three configurations lead to a maximum thrust available of 1.1970 N, 0.8550 N and 0.3420 N. As expected, there is no difference in the slow solutions, even due to the maximum mission time duration set to 2 years. The solutions with low propellant consumption and high mission duration are characterized by the choice of a a_d really similar to the semi-major axis of the orbits of the constellation, solutions that can be carried out with each propulsive configuration. The real difference is in the fastest solutions, since the higher the maximum thrust available the faster an orbit with different a_d can be reached. Ten thrusters allow the deployment of the whole set of satellites in about 100 days, while two thrusters only require at least about 250 days to release the six of them.

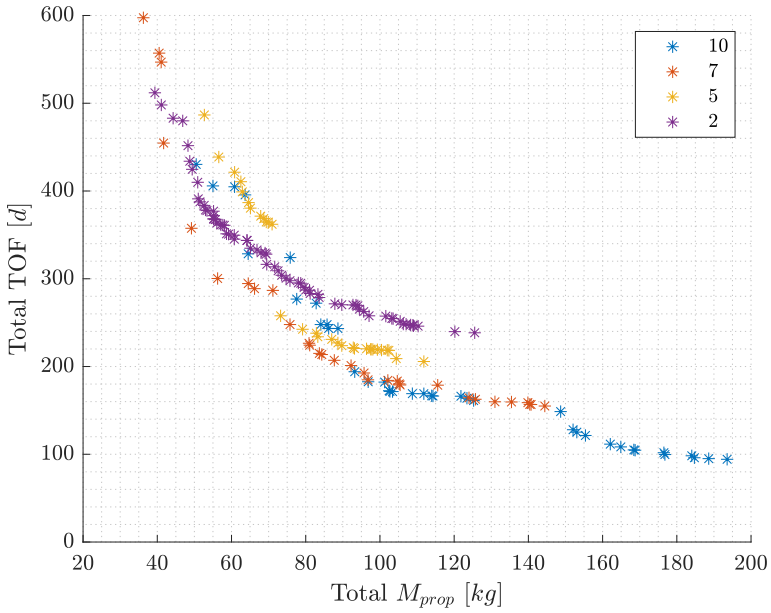


Figure 5.2: Pareto fronts with different number of thrusters for Starlink replacement mission ($M_0 = 2000\ kg$; $I_{sp} = 3500\ s$).

5.2.2 Nano-satellites constellation deployment

It is interesting to analyze the case in which a complete constellation is deployed in one launch thanks to the multi-deployment strategy developed in this work. A constellation of nano-satellites is considered in order to keep the overall dry mass low. In particular, three different cases of the same problem have been investigated.

1. **Case 1**: Walker constellation ($N = 8$), 1 satellite per each orbit.
2. **Case 2**: Walker constellation ($N = 8$), 5 satellites per each orbit.
3. **Case 3**: Walker constellation ($N = 16$), 1 satellite per each orbit.

Case 1 A Walker constellation with $N = 8$ circular orbits with 53° inclination and an altitude of 1000 km was considered. The KP of the orbits are summarised in Table 5.5. Each orbit belongs to an orbital plane which is 45°

a [DU]	e [-]	i [deg]
1.1568	0	53

Table 5.5: KP of Walker constellation.

separated from the two adjacent planes. One nano-satellite of 10 kg has to be released on each orbit. According to eq. (5.1), the dry mass of the vehicle would be $M_v = 136.29\ kg$. The wet mass in this case was rounded up to 200 kg. The propulsive characteristics of the vehicle are the ones explained in section 5.1. This time, the vehicle was considered to be provided with two RIT 2X thrusters, for a total maximum thrust available of 0.342 N. The case was anaysed with a subsearch dimension of $r = 4$, therefore requiring two iterations of the hybrid optimization method to find the final solutions. Again, different tuning parameters, listed in Table 5.6, were selected for the

	Iter 1	Iter 2
N_p	20	15
M_{gen}	40	20

Table 5.6: Optimization tuning parameters for nano-satellites constellation deployment case 1.

two iterations due to the differences in the size of the search space. The lb and ub of the variables are instead presented in Table 5.7.

Parameter	lb	ub
a_d [DU]	1.0314	1.5000
i_d [deg]	53	53

Table 5.7: Lower and upper bounds for nano-satellites constellation deployment case 1.

The results of the optimization are presented in Figure 5.3. This multi-deployment strategy would allow to release a whole constellation with only one launch and in less than one year with a contained amount of propellant consumption. As expected, due to the particular configuration of the releasing orbits, the best releasing order is the one which includes releasing the satellites on each orbit in order of RAAN. In particular, all the solutions of the Pareto in Figure 5.3 belong to the same releasing order $p = [5, 6, 7, 8, 1, 2, 3, 4]$, found by the algorithm to be the optimal one.

Case 2 Up to now, only cases in which only one satellite has to be released on one orbit have been considered. The real potential of this multi-deployment strategy can be seen when a cluster of satellites has to be released on the same orbit with a desired phasing difference. The same case was therefore analyzed but this time with 5 satellites of 10 kg to release on each of the 8 orbits. To have the satellites equidistant from the others in

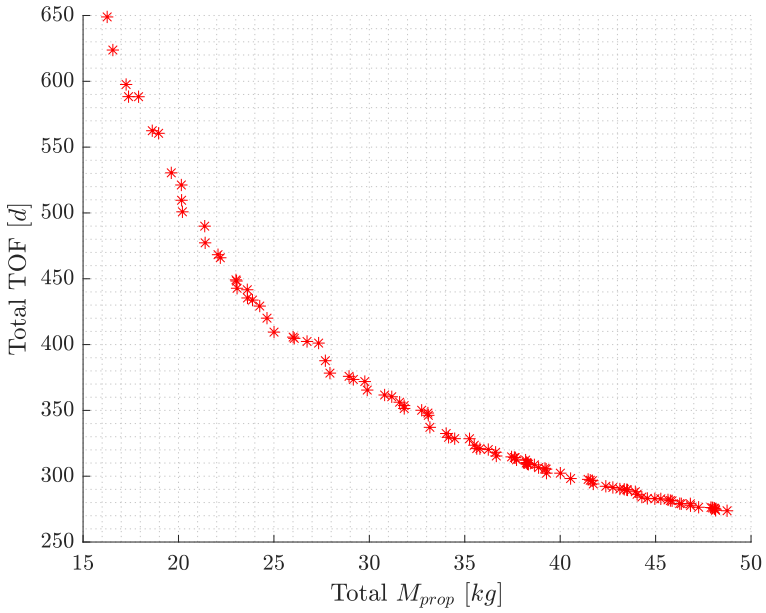


Figure 5.3: Pareto front for nano-satellites constellation deployment case 1 ($M_0 = 200\ kg$; $I_{sp} = 3500\ s$).

the same orbit, there must be a 72° gap in anomaly between each of them. According to eq. (5.1), the dry mass of the vehicle would be $M_v = 480.45$ kg. The wet mass, in this case, was rounded up to 700 kg. The propulsive characteristics of the vehicle are the ones explained in section 5.1. The vehicle was considered to be provided with five RIT 2X thrusters, for a total maximum thrust available of 0.855 N. The optimization was performed as before, again with the tuning parameters in Table 5.6 and the lower and upper bounds in Table 5.7.

The phasing maneuver is computed through the approach explained in section 4.3. The values of the semi-major axis a_2 to reach is chosen iteratively by changing the number of revolutions N_{rev} to wait on the orbit, such that $a_2 > 0.97 \cdot a_1$. All the orbits are at 1000 km altitude, therefore being characterized by $a_1 = 1.1568$ DU. The consequent phasing orbit will have $a_2 = 1.1257$ DU, with a number of revolutions to wait on that orbit $Nrev = 5$ to perform the 72° phasing maneuver. This value is the same for all the phasing maneuvers to perform since its computation only depends on the amount of phasing needed and on a_1.

The results are visible in Figure 5.4. While the impulsive approximation grants a good accuracy on the propellant cost, it may slightly underestimate the duration of the phasing maneuver. However, the latter is about two orders of magnitude lower than the total time of the mission and is

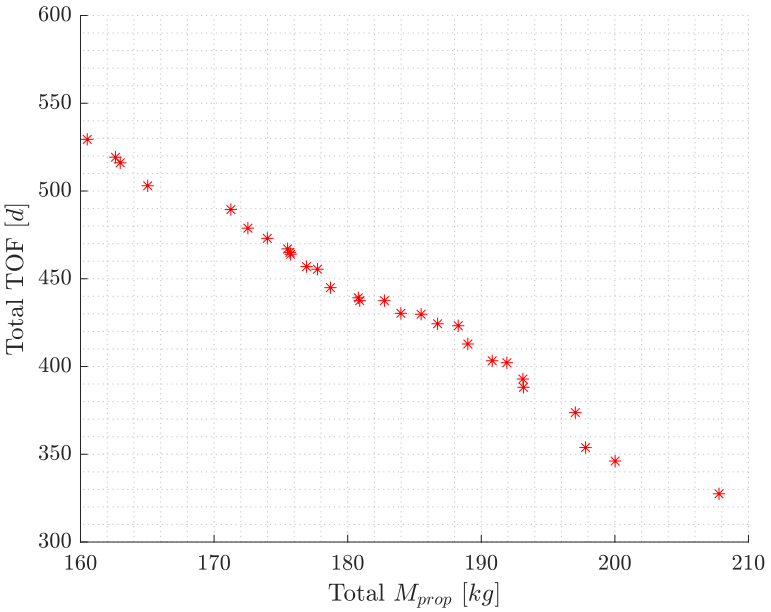


Figure 5.4: Pareto front for nano-satellites constellation deployment case 2 ($M_0 = 700\ kg$; $I_{sp} = 3500\ s$).

therefore a valid approximation at this point of the analysis. The propellant consumption of course increases with respect to Figure 5.3, since the phasing maneuvers have of course a cost and also because the initial mass of the spacecraft is much higher than in the previous case. However, the total time of the mission shows how it is possible in about the same amount of time to release a huge number of satellites at the desired state. The case of constellation deployment is the classical application in which more than one satellite has to be released on the same orbit. However, independently from the final mission of the satellites, the results in Figure 5.4 show how clustering the satellites to launch in such a way that more than one has to be released on the same orbit would make the low-thrust multi-deployment strategy extremely competitive.

Case 3 The case of the deployment of a nano-satellite constellation was also considered the perfect one to finally study a problem with size $N > 10$. The same problem as in case 1 was therefore considered for a $N = 16$ problem to test the performance and quality of results of the hybrid optimization method. The constellation this time is characterized by 16 orbital planes shifted of 22.5° in RAAN one from another. Again, only one spacecraft to be released on each orbit was considered. In this case, the dry mass of the vehicle would be $M_v = 222.3\ kg$ and therefore the wet mass was rounded up

to 300 kg. The same number of thrusters as before was kept for the analysis. The optimization was performed again with a sub-search size $r = 4$. Again, the tuning parameters of the optimization at the various steps were different, due to the different sizes of the sub-problems. In particular, a high value of N_p and M_{gen} were necessary for the first iteration since the the matrix $\mathbf{P}$, according to eq. (2.6), contains $N_{perms} = 43680$ possible releasing orders. The selected values are visible in Table 5.8.

	Iter 1	Iter 2	Iter 3	Iter 4
N_p	30	25	20	15
M_{gen}	50	40	30	20

Table 5.8: Optimization tuning parameters for nano-satellites constellation deployment case 3.

The Pareto front resulting from the latter is represented in Figure 5.5. At least one year is required to complete the release of all the satellites on board. All the solutions found share the same first 8 elements of the releasing order ($p = [13, 14, 15, 16, 4, 3, 2, 1]$), while changing the release orders of the last 8. This feature is due to the nature of the search which is heuristic even though with a branch-and-bound pruning iteration by iteration.

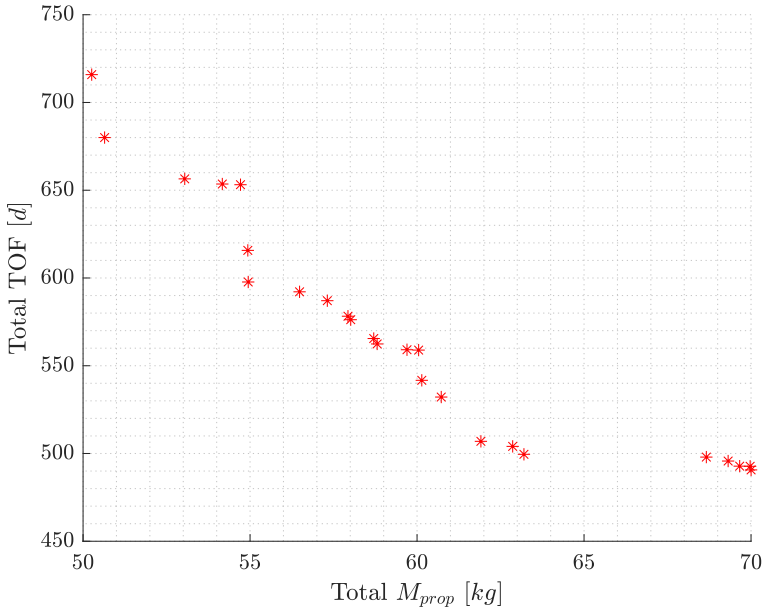


Figure 5.5: Pareto front for nano-satellites constellation deployment case 3 ($M_0 = 300\ kg$; $I_{sp} = 3500\ s$).

5.3 Multi-mission insertion

In this section, two examples involving the deployment of satellites belonging to different missions are reported. First, an example of the deployment of different satellites onto SSOs is considered. Afterward, a special application is presented, where the algorithm is used not only to select the transfer strategy but also to select the satellites to release.

5.3.1 SSO satellites multi-deployment

Most of the satellites launched nowadays are designed to be released on SSOs. For this reason, this class of orbits was chosen for this case of multi-mission deployment. Nine different satellites to be released on nine different orbits were considered. SSOs are commonly referred to through the LTAN parameter, rather than the RAAN. The orbits are listed in Table 5.9. Orbits

ID	a [DU]	e [-]	i [deg]	LTAN [h]
0	1.1000	0	98.60	12.00
1	1.1087	0	98.60	12.30
2	1.1277	0	98.65	12.00
3	1.1171	0	98.50	11.30
4	1.1152	0	98.45	11.30
5	1.1226	0	98.65	12.00
6	1.1189	0	98.55	11.30
7	1.1201	0	98.58	11.30
8	1.1239	0	98.66	12.00
9	1.1167	0	98.49	12.00

Table 5.9: SSO satellites multi-deployment initial states.

1 and 2 are respectively the orbits onto which Sentinel-1 [44] and Sentinel-3 [45] are released, apart from the LTAN which was chosen arbitrarily. Semi-major axis and inclination of the other orbits in Table 5.9 were fixed such to have a nodal regression rate $\dot{\Omega}_{sec}$ due to J2 equal to 1 degree per day, which is approximately the regression needed to have a SSO. Fixing the inclination i and $\dot{\Omega}_{sec}$, it was possible to find the value of the semi-major axis a of the orbit from eq. (2.3). Through the use of root-finding algorithms, it was possible to locate the value of a granting the correct nodal regression since both n and p depend on such value, as it is shown in eq. (5.2) where μ represents the standard gravitational parameter of the planet.

$$p = a \cdot (1 - e^2) \tag{5.2a}$$

$$n = \sqrt{\frac{\mu}{a^3}} \tag{5.2b}$$

The values of LTAN were chosen to be similar from one to another in order to have final results with reasonable duration and propellant consumption. While in the previous examples (section 5.2) large differences in RAAN were accounted for, this time orbits belonging to the same region of space were selected due to their inclination value. As it is clear from eq. (2.3), the higher the inclination the slower the nodal regression due to J2. Large RAAN changes when dealing with nearly-polar orbits would require either a really large amount of time (in the order of several months or even years) or large plane change maneuvers which would accelerate the nodal regression but also dramatically increase the propellant consumption. For these reasons, the most likely application for a multi-mission deployment in this region of space around Earth was considered to be the deployment of several small-satellites on different orbits but characterized by similar values of inclination and LTAN, like the ones in Table 5.9.

The mass of the satellites to release are listed in Table 5.10. Differently

ID	1	2	3	4	5	6	7	8	9
Mass $[kg]$	114	125	50	60	50	40	10	8	9

Table 5.10: SSO satellites multi-deployment mass.

from the previous example, this time the masses of the satellites are different from each other since each of them belongs to a different mission and therefore has different purposes. The masses were chosen arbitrarily to have an assorted cluster of satellites: two small-satellites ($> 100\ kg$), four micro-satellites (10-100 kg), three nano-satellites (1-10 kg). Dealing with satellites with different mass increases the complexity of the releasing order definition. As it is known, the cost of a transfer depends on the distance of the target but also on the mass of the vehicle. While previously there was no difference in releasing one satellite instead of another in terms of mass, in this case releasing the heavier satellites first would also deeply impact the mass of propellant needed for the successive transfers.

By using eq. (5.1) it was possible to compute $M_v = 551.4\ kg$. Considering the propellant, the initial mass M_0 was rounded up to 700 kg. Five RIT 2X thrusters were considered for a preliminary analysis.

The problem has $N = 9$ orbits to reach and was solved considering a sub-search of dimension $r = 3$. Therefore, three iterations of the hybrid optimization approach were necessary to solve the problem and the tuning parameters of each iterations are shown in Table 5.11. Lower and upper bounds are reported in Table 5.12.

	Iter 1	Iter 2	Iter 3
N_p	20	20	15
M_{gen}	50	30	25

Table 5.11: Optimization tuning parameters for SSO satellites multi-deployment.

Parameter	lb	ub
a_d [DU]	1.0314	1.6000
i_d [deg]	97	103

Table 5.12: Lower and upper bounds for SSO satellites multi-deployment.

Usually, the search space of i_d is defined between the minimum and the maximum inclinations of the orbits to reach. In this case, since the mission copes with nearly-polar orbits, the search space is enlarged in order to give the possibility to accelerate the nodal regression at the cost of a higher amount of propellant consumption to change the plane.

The results of the optimization are shown in Figure 5.6. The minimum

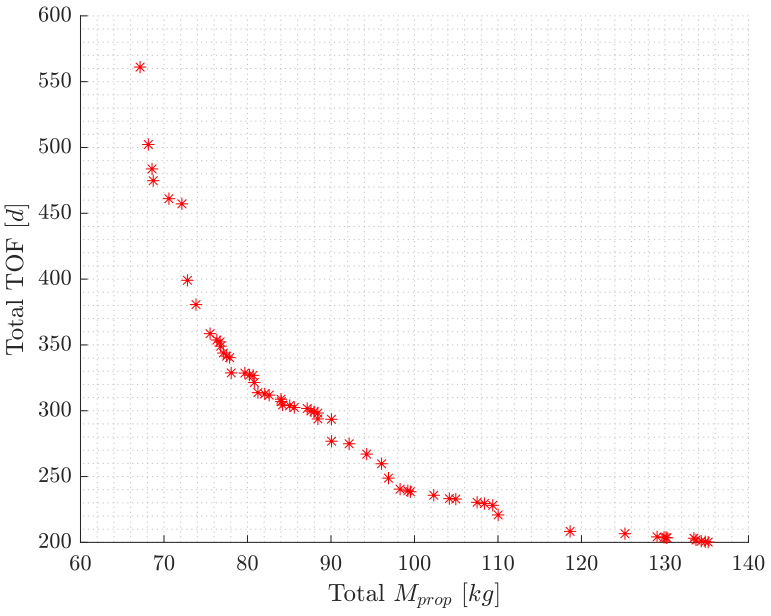


Figure 5.6: Pareto front for SSO satellites multi-deployment ($M_0 = 700\ kg$; $I_{sp} = 3500\ s$).

time found to release all the satellites is around 200 days, while, again, the maximum mission time was set to be 2 years. It is interesting to focus on the releasing orders of the optimal solutions. As aforementioned, when some satellites are much heavier than others it is convenient to release them as soon as possible to minimize the overall propellant consumption. This is confirmed by the releasing orders of the solutions found since satellites 1 and 2 are always two of the first three satellites released. 10 different releasing orders were found as optimal solutions, even if not highlighted in different colors in Figure 5.6 this time.

It is interesting also in this case to study the same problem with a different amount of thrusters. The previous results are compared to the case in which the vehicles is provided with only two electic thrusters. The results of the comparison are shown in Figure 5.7. Again, it can be seen how the

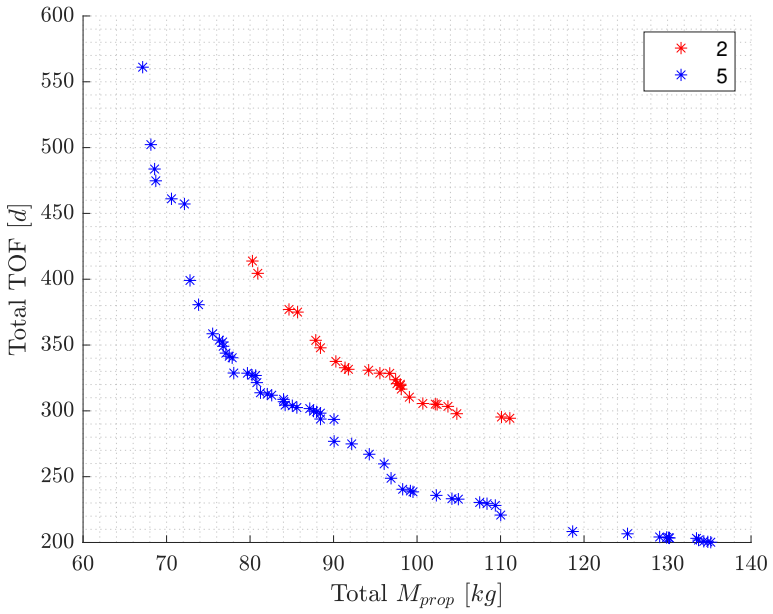


Figure 5.7: Pareto fronts comparison for SSO satellites multi-deployment ($M_0 = 700\ kg$; $I_{sp} = 3500\ s$).

satellites can be released in reasonable amounts of time even with a lower thrust authority. This is due to the fact that the greatest contribution to the total mission duration is given by the stationary time waited on the drifting orbits rather than the TOF of the low-thrust transfers.

5.3.2 Satellites clustering choice

The last case analyzed is slightly different from the previous ones. This time, the multi-deployment algorithm is used not only to optimize the deploying

strategy in terms of propellant and duration but also to choose which satellites to deploy with that particular launch. Given a set of N satellites, the algorithm this time has to define the best sub-sets of dimension n, with $n < N$.

This approach is applied to a multi-mission deployment case. Also, this was the chance to test the algorithm when working with orbits with larger differences in the semi-major axis. A set of $N = 9$ orbits was selected, whose parameters are shown in Table 5.13. A satellite of 50 kg was considered to be released on each orbit. The objective of the algorithm is therefore finding the best subsets, of dimension $n = 6$, to deploy.

ID	a [AU]	e [-]	i [deg]	Ω [deg]
0	1.1000	0	98.60	281.37
1	1.1087	0	98.60	285.87
2	1.1277	0	98.65	281.37
3	1.1171	0	98.50	270.87
4	1.1152	0	98.45	270.87
5	1.4000	0	98.65	200.00
6	1.3000	0	98.55	50.00
7	1.2000	0	98.58	70.00
8	1.2500	0	98.66	6.00
9	1.3700	0	98.49	134.00

Table 5.13: Satellites clustering choice initial states.

Considering that the vehicle will have on-board n satellites, according to eq. (5.1) $M_v = 372.9\ kg$ was found. Considering the propellant, the initial mass M_0 was rounded up to 500 kg. Five RIT 2X thrusters were considered for a preliminary analysis.

The lower and upper bounds of the optimization are shown in Table 5.14. The optimization was performed through the use of the hybrid approach,

Parameter	lb	ub
a_d [DU]	1.0314	1.600
i_d [deg]	97	103

Table 5.14: Lower and upper bounds for satellites clustering choice.

with a sub-search dimension $r = 3$ but stopping after two iterations, such to achieve a total $N = 6$. The tuning parameters of the two are shown in Table 5.15.

The resulting Pareto front is represented in Figure 5.8 and the releasing

	Iter 1	Iter 2
N_p	20	10
M_{gen}	30	20

Table 5.15: Optimization tuning parameters for satellites clustering choice.

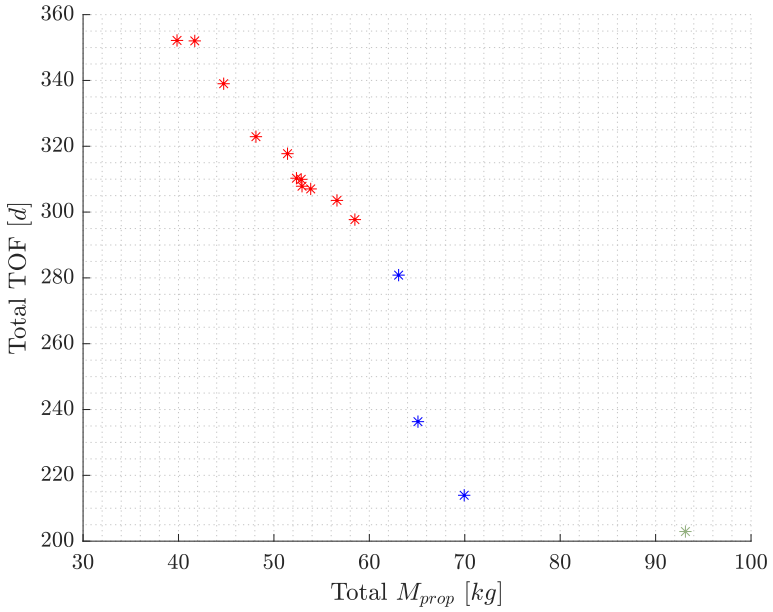


Figure 5.8: Pareto front for satellites clustering choice ($M_0 = 500\ kg$; $I_{sp} = 3500\ s$).

orders of the subset belonging to the optimal solutions are listed in Table 5.16. Three different releasing orders were found to be optimal but only two clusters of satellites, since the red and blue solutions involve the same satellites but with a different order of deployment.

It can be observed that the first three releases are the same for all of them, which could be expected given the similarities to the initial state KP_0. Since also the orbit 4 is really close to the first three, all the releasing orders but one consider deploying this satellite immediately after having released the first three on orbits 1, 2 and 3. Differently, this run of the optimization found a solution that does not consider the deployment of the satellite 4, preferring other satellites to deploy, to achieve a faster overall deployment at the cost of an increase in propellant consumption.

Even if presented as last, this application of the multi-deployment algorithm could turn really useful when planning not only the deploying strategy

Color	Releasing order
Red	[3,2,1,4,6,8]
Blue	[3,2,1,4,8,6]
Green	[3,2,1,8,7,9]

Table 5.16: Releasing orders for satellites clustering choice.

but also the launches. This application can indeed suggest, out of a large set of satellites, which are the most convenient ones to put to orbit in a single launch and, in addition to this, the deploying strategy to follow to insert them.

6. Conclusions

THE main goal of this work of thesis, which is the development of a computationally fast algorithm for the planning of a multi-deployment mission, has been achieved.

First, a competitive routing algorithm, whose performances showed really good results in the comparison with the GTOC problem, was developed. Besides its promising performances, the great advantage of the developed routing algorithm is in the numerous fields of applications it can be applied to. While this thesis focuses on the analysis of a multi-deployment mission scenario, the routing algorithm presented in chapter 2 can be applied to any routing problem, such as ADR, MGA or on-orbit servicing.

In addition to this, a branch and bound based heuristic approach was developed and used for the optimization. Since MINLPs belong to the NP-hard problems family, studying the whole routing problem at once becomes the most important challenge when dealing with problems of large dimensions. This hybrid optimization approach which divides the problem into various subsets and performs several consequential heuristic optimizations proved to achieve better results than a pure heuristic approach which directly addresses the whole routing problem.

In addition to these considerations, some comments on the results found in the simulations in chapter 5 must be done. This thesis particularly focused on a dispenser with a low-thrust control authority. This choice was driven by the analysis in chapter 4 which highlighted how performing the transfer strategy selected with impulsive maneuvers would lead to way higher propellant consumptions, not justified by a decrease in the mission duration. The choice of the low-thrust propulsive system and above all of the J_2 exploiting transfer strategy lead to total multi-deployment mission durations in the order of months. These mission durations are indeed justified by the choice of the orbits into which to release the satellites. It was chosen to address the problem focusing on orbits with very different RAAN values, a choice that required to plan the transfers exploiting the J_2 effect to make a mission with multiple transfers feasible. While the order of magnitude of the mission duration may sound unattractive, the alternatives to accomplish the deployment of satellites with such RAAN differences must be analyzed. A

first alternative may be to launch them in separate launches, deploying each of them in the correct region in space. This option would of course drastically reduce the time to operations of the satellites. However, it may be difficult to arrange the launches when several satellites need to be released each in a different region. Alternatively, a single launch solution may be to focus on impulsive maneuvers and to not exploit the J_2 effect to achieve the RAAN changes. This would of course decrease the duration of the single transfers but would also make the propellant consumption increase to infeasible amounts.

Finally, some possible future works on the problem are here suggested:

- The introduction of the J2 effect punctually on the orbit, rather than only considering the secular effect, could be considered. This would allow to more carefully design the trajectories and consequently to get more accurate results in terms of control low, propellant consumption and times of flight. However, this should be achieved through an analytical or semi-analytical approach to not increase the computational load of the algorithm.

- One other extension of the algorithm may also be addressing more in detail the case of several satellites to release on one orbit. In particular, it should be studied how the optimization of the phasing maneuver could be integrated with the optimization of the multi-deployment mission and in particular of the releasing order.

- Finally, it would be interesting to study the problem from a six degrees of freedom point of view, paying attention to the attitude of the releasing vehicle. In particular, the configuration of the vehicle and how the releases affect its inertia values should be investigated.

Bibliography

[1] SpaceWorks. *2020 Nano/Microsatellite Market Forecast, 10th Edition*. 2020 (cit. on pp. i, iii, 1).

[2] Euroconsult. *Satellites To Be Built and Launched by 2028. A complete analysis & forecast of satellite manufacturing & launch services*. 2019, pp. 5–10 (cit. on pp. i, iii, 1).

[3] Bradley J. Wall and Bruce A. Conway. "Shape-based approach to low-thrust rendezvous trajectory design". In: *Journal of Guidance, Control, and Dynamics* 32.1 (2009), pp. 95–102. ISSN: 15333884. DOI: 10.2514/1.36848 (cit. on pp. i, iii, 6).

[4] Jin Zhang et al. "Analysis of multiple asteroids rendezvous optimization using genetic algorithms". In: *2015 IEEE Congress on Evolutionary Computation, CEC 2015 - Proceedings* (2015), pp. 596–602. DOI: 10.1109/CEC.2015.7256945 (cit. on pp. i, iii, 4, 28, 29).

[5] Stefano Silvestrini et al. "Design of Robust Passively Safe Relative Trajectories for Uncooperative Debris Imaging in Preparation to Removal". In: *2020 AAS/AIAA Astrodynamics Specialist Conference*. 2020, pp. 1–18 (cit. on pp. i, iii, 22).

[6] NASA. *What are SmallSats and CubeSats?* URL: https://www.nasa.gov/content/what-are-smallsats-and-cubesats. //accessed: 30.03.2021 (cit. on p. 1).

[7] Moog Inc. *SL-OMV Datasheet*. 2019. URL: https://www.moog.com/content/dam/moog/literature/Space_Defense/spaceliterature/omv/Moog-SLOMV-Datasheet.pdf. //accessed: 30.03.2021 (cit. on p. 1).

[8] Spaceflight. *Sherpa Program: New Orbital Transfer Vehicles Launch Smallsats to Custom Orbital Destinations*. URL: https://spaceflight.com/sherpa/. //accessed: 30.03.2021 (cit. on p. 1).

[9] D-Orbit. *ION CubeSat Carrier*. URL: https://www.dorbit.space/inorbit-now. //accessed: 30.03.2021 (cit. on p. 1).

[10] Brent William Barbee et al. “Design of spacecraft missions to remove multiple orbital debris objects”. In: *2011 Aerospace Conference*. IEEE. 2011, pp. 1–14 (cit. on p. 2).

[11] Nicolas Bérend and Xavier Olive. “Bi-objective optimization of a multiple-target active debris removal mission”. In: *Acta Astronautica* 122 (2016), pp. 324–335. ISSN: 00945765. DOI: 10.1016/j.actaastro.2016.02.005 (cit. on pp. 2, 3).

[12] Dario Izzo et al. “Evolving solutions to TSP variants for active space debris removal”. In: *GECCO 2015 - Proceedings of the 2015 Genetic and Evolutionary Computation Conference* (2015), pp. 1207–1214. DOI: 10.1145/2739480.2754727 (cit. on pp. 2, 3).

[13] Jun Bang and Jaemyung Ahn. “Multitarget Rendezvous for Active Debris Removal Using Multiple Spacecraft”. In: *Journal of Spacecraft and Rockets* 56.4 (2019), pp. 1237–1247 (cit. on p. 2).

[14] Kyle T. Alfriend, Deok Jin Lee, and N. Glenn Creamer. “Optimal servicing of geosynchronous satellites”. In: *AIAA/AAS Astrodynamics Specialist Conference and Exhibit* August (2002), pp. 1–9. DOI: 10.2514/6.2002-4905 (cit. on pp. 2, 3).

[15] Jin Zhang et al. “Multispacecraft refueling optimization considering the J 2 perturbation and window constraints”. In: *Journal of Guidance, Control, and Dynamics* 37.1 (2014), pp. 111–122 (cit. on p. 2).

[16] Paolo De Pascale and Massimiliano Vasile. “Preliminary design of low-thrust multiple gravity-assist trajectories”. In: *Journal of Spacecraft and Rockets* 43.5 (2006), pp. 1065–1076 (cit. on p. 2).

[17] C. H. Yam, D. Izzo, and D. D. Lorenzo. “Low-thrust trajectory design as a constrained global optimization problem”. In: *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering* 225.11 (2011), pp. 1243–1251. ISSN: 20413025. DOI: 10.1177/0954410011401686 (cit. on p. 3).

[18] Bradley J. Wall and Bruce A. Conway. “Genetic algorithms applied to the solution of hybrid optimal control problems in astrodynamics”. In: *Journal of Global Optimization* 44.4 (2009), pp. 493–508. ISSN: 09255001. DOI: 10.1007/s10898-008-9352-4 (cit. on p. 4).

[19] Judea Pearl. *Heuristics Intelligent Search Strategies for Computer Problem Solving*. Addison-Wesley Publishing Company, 1984, pp. 3–4 (cit. on p. 4).

[20] Christian Blum and Andrea Roli. “Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison”. In: *ACM Computing Surveys* 35.3 (2003), pp. 268–308. ISSN: 03600300. DOI: 10.1145/937503.937505 (cit. on pp. 4, 5).

[21] J. Kennedy and R. Eberhart. "Particle swarm optimization". In: *Proceedings of ICNN'95 - International Conference on Neural Networks* 4 (1995), 1942–1948 vol.4. DOI: 10.1109/ICNN.1995.488968 (cit. on p. 5).

[22] John Henry Holland et al. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, 1992 (cit. on p. 5).

[23] Rafael Martí, Panos M. Pardalos, and Mauricio G.C. Resende. *Handbook of heuristics*. Vol. 1-2. 2018, pp. 1–1385. ISBN: 9783319071244. DOI: 10.1007/978-3-319-07124-4 (cit. on p. 5).

[24] Víctor Martínez-Cagigal. *Multi-Objective Particle Swarm Optimization (MOPSO)*. MATLAB Central File Exchange. 2020. URL: https://www.mathworks.com/matlabcentral/fileexchange/62074-multi-objective-particle-swarm-optimization-mopso (cit. on pp. 5, 14).

[25] Howard D Curtis. *Orbital mechanics for engineering students*. Butterworth-Heinemann, 2013, pp. 59–79 (cit. on p. 5).

[26] HS Tsien. "Take-off from satellite orbit". In: *Journal of the American Rocket Society* 23.4 (1953), pp. 233–236 (cit. on p. 6).

[27] Richard H Battin. *An Introduction to the Mathematics and Methods of Astrodynamics, revised edition*. American Institute of Aeronautics and Astronautics, 1999 (cit. on p. 6).

[28] John T. Betts. "Very low-thrust trajectory optimization using a direct SQP method". In: *Journal of Computational and Applied Mathematics* 120.1 (2000), pp. 27–40. ISSN: 03770427. DOI: 10.1016/S0377-0427(00)00301-0 (cit. on p. 6).

[29] Craig A. Kluever and Steven R. Oleson. "Direct approach for computing near-optimal low-thrust earth-orbit transfers". In: *Journal of Spacecraft and Rockets* 35.4 (1998), pp. 509–515. ISSN: 15336794. DOI: 10.2514/2.3360 (cit. on p. 6).

[30] Bruce A Conway. *Spacecraft trajectory optimization*. Vol. 29. Cambridge University Press, 2010 (cit. on p. 6).

[31] Anastassios E Petropoulos and Jon a Sims. "A Review of Some Exact Solutions to the Planar Equations of Motion of a Thrusting Spacecraft Anastassios". In: *International Symposium Low Thrust Trajectories* 2.1 (2002), pp. 1–14 (cit. on p. 6).

[32] Bradley J. Wall. "Shape-based approximation method for low-thrust trajectory optimization". In: *AIAA/AAS Astrodynamics Specialist Conference and Exhibit* August (2008), pp. 1–9. DOI: 10.2514/6.2008-6616 (cit. on p. 6).

[33] Jacopo Prinetto and Michelle Lavagna. "Elliptical shape-based model for multi-revolution planeto-centric mission scenarios". In: *Celestial Mechanics and Dynamical Astronomy* 133.1 (2021), pp. 1–24. ISSN: 15729478. DOI: 10.1007/s10569-020-10001-9 (cit. on pp. 6, 34).

[34] Howard D Curtis. *Orbital mechanics for engineering students*. Butterworth-Heinemann, 2013, pp. 247–258 (cit. on p. 22).

[35] Howard D Curtis. *Orbital mechanics for engineering students*. Butterworth-Heinemann, 2013, pp. 312–316 (cit. on p. 27).

[36] Jun Bang and Jaemyung Ahn. "Two-phase framework for near-optimal multi-target Lambert rendezvous". In: *Advances in Space Research* 61.5 (2018), pp. 1273–1285. ISSN: 18791948. DOI: 10.1016/j.asr.2017.12.025 (cit. on pp. 28, 30).

[37] MATLAB *Optimization Toolbox*. The MathWorks, Natick, MA, USA. 2020b. URL: https://it.mathworks.com/products/global-optimization.html (cit. on p. 29).

[38] P. Fortescue, G. Swinerd, and J. Stark. *Spacecraft Systems Engineering*. Wiley, 2011. ISBN: 9781119978367 (cit. on p. 39).

[39] Haibin Shang, Shuai Wang, and Weiren Wu. "Design and optimization of low-thrust orbital phasing maneuver". In: *Aerospace Science and Technology* 42 (2015), pp. 365–375. ISSN: 12709638. DOI: 10.1016/j.ast.2015.02.003 (cit. on pp. 49, 50).

[40] Wiley J Larson and James Richard Wertz. *Space mission analysis and design*. Tech. rep. Torrance, CA (United States); Microcosm, Inc., 1992 (cit. on p. 59).

[41] George P Sutton and Oscar Biblarz. *Rocket propulsion elements*. John Wiley & Sons, 2016, p. 660. ISBN: 9781118753880 (cit. on p. 60).

[42] ArieneGroup. *Electric Ion Space Propulsion Systems and Thrusters*. URL: https://www.space-propulsion.com/spacecraft-propulsion/propulsion-systems/electric-propulsion/index.html. //accessed: 30.03.2021 (cit. on p. 60).

[43] ESA. *Starlink Satellite Constellation of SpaceX*. URL: https://directory.eoportal.org/web/eoportal/satellite-missions/s/starlink. //accessed: 30.03.2021 (cit. on p. 60).

[44] ESA. *Sentinel-1*. URL: https://sentinel.esa.int/web/sentinel/missions/sentinel-1/satellite-description/orbit. //accessed: 30.03.2021 (cit. on p. 69).

[45] ESA. *Sentinel-3*. URL: https://sentinel.esa.int/web/sentinel/missions/sentinel-3/satellite-description/orbit. //accessed: 30.03.2021 (cit. on p. 69).