Trabajo Fin de Grado
Ingeniería Aeroespacial

An Ant-Colony Based Optimization Algorithm for Interplanetary Trajectories

Autor: Juan Antonio Villajos Molina

Tutor: Rafael Vázquez Valenzuela, Julio César Sánchez Merino

Dpto. de Ingeniería Aeroespacial y Mecánica de Fluidos
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2020

Trabajo Fin de Grado
Ingeniería Aeroespacial

An Ant-Colony Based Optimization Algorithm for Interplanetary Trajectories

Autor:
Juan Antonio Villajos Molina

Tutor:
Rafael Vázquez Valenzuela, Julio César Sánchez Merino
Profesor Titular, Contrato Predoctoral

Dpto. de Ingeniería Aeroespacial y Mecánica de Fluidos
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2020

Trabajo Fin de Grado: An Ant-Colony Based Optimization Algorithm for Interplanetary Trajectories

Autor: Juan Antonio Villajos Molina
Tutor: Rafael Vázquez Valenzuela, Julio César Sánchez Merino

El tribunal nombrado para juzgar el trabajo arriba indicado, compuesto por los siguientes profesores:

Presidente:

Vocal/es:

Secretario:

acuerdan otorgarle la calificación de:

El Secretario del Tribunal

Fecha:

Contents

Nomenclature

$\odot$	The Sun
$\oplus$	The Earth
$\male$	Mars
$\saturn$	Saturn
$\jupiter$	Jupiter
$\female$	Venus
a	Semi major axis
e	Eccentricity vector
i	Inclination
θ	True anomaly
ω	Argument of periapsis
Ω	Right ascension of the ascending node
h	Specific angular momentum
μ	Gravitational parameter
ε	Specific energy
n	Node line
$\mathbf{C}_{\mathrm{H}}^{\mathrm{F}}$	Rotation matrix from Perifocal to Heliocentric frame of reference
L	Mean longitude
M	Mean anomaly
E	Eccentric anomaly
R_P	Planetary radius
L_P	Mean distance from the Sun to planet P
R_{e_P}	Radius of the Sphere of Influence of planet P
h_{park}	Altitude of the parking orbit
V_∞ , V^P	Hyperbolic excess speed
C_3	Mission characteristic energy
r_p	Periapse radius
δ	Turn angle of the asymptote of the hyperbola
T_{syn}	Synodic period
GAM	Gravity Assist Maneuver
DSM	Deep Space Maneuver
MGA	Multiple Gravity Assist Maneuvers

1 Introduction

1.1 Problem Statement

Humanity's interest in the heavens has led to space exploration, which helps to expand technology, create new industries and foster the links between nations. As an outgrowth of the mid-20th-century Cold War, the Space Race began. It was a series of competitive technology demonstrations between the United States and the Soviet Union, aiming to show superiority in spaceflight. Although the last years of the Space Race were focused on reaching the Moon, it is worthwhile to mention the effort made by the Soviet Union to visit other planets. In fact, in 1965, Venera 3 became the first spacecraft to land on another planet's surface (Venus).

After that, the number of planetary missions quickly increased. Some of the most notable mission examples for the targets under consideration in this project are:

- Mars: Mars 2 (launched in 1971) represents the first man-made object on Mars, Mariner 9 (launched in 1971) became the first spacecraft to orbit another planet, Viking 1 (launched in 1975) was the first successful landing on Mars and Mars Pathfinder spacecraft (launched in 1996) became the first rover to operate on another planet.

- Saturn: Pioneer 11 (launched in 1973) intercepted Saturn after a flyby of Jupiter, taking the first ever close up pictures of the Ringed Planet; Voyager 1 (launched in 1977) flew within 124000 km of Saturn and also made a flyby of its moon Titan; Voyager 2 (launched in 1977) made flybys of other Saturn's moons such as Enceladus or Tethys; and Cassini-Huygens (launched in 1997) successfully went into orbit around Saturn.

- Jupiter: Galileo (launched in 1989) was the first mission sent to orbit Jupiter, Ulysses (launched in 1990) measured Jupiter's intense magnetic and radiation field with a flyby, and New Horizons' (launched in 2006) flyby of Jupiter was essential to reach Pluto.

In the past several decades, serious progress has been made in the design of interplanetary missions. Not only the technology used, but also the methods employed to reach other planets have been amazingly developed. For instance, the concept of Gravity Assist Maneuvers made possible the access to high cost targets in the Solar System, with significant reductions in launch energy. Consequently, multiple gravity assist (MGA) trajectories have been extensively investigated over the last decades. Deep Space Maneuvers frequently improve the performance of space missions indeed, decreasing the propellant cost.

However, the design of interplanetary transfers is a highly non-linear problem which involves myriad optimization parameters. Hence it is usually a non-convex problem which leads to multiple local minimums in the search space. Accordingly, a challenging difficulty which arises when solving interplanetary trajectories is finding the global optimal solution. The intensive exploration of the search space is essential to discover assorted solutions, in order to compare them and select the optimal one. Thus, an algorithm which does not get stuck in local minimums is required.

1.2 Scope of the Project

Global optimization approaches have been broadly used towards the solution of complex interplanetary transfers. Commonly, Metaheuristic-based routines are used to tackle this problem. However, the efficiency, both computational and performance-wise, of these methods are strongly linked to the type of problem under study. To give an illustration, the use of Deep Space Maneuvers and Gravity Assist Maneuvers brings a mathematical challenge as the dimensionality of the problem increases. Nonetheless, these stochastic global techniques, such as genetic algorithms, are usually very expensive in terms of computation time, often taking a prohibitive amount of time to compute an optimal solution.

In this project, an alternative approach is used to tackle the problem. The local optimization solver *fmincon*, provided from Matlab Optimization Toolbox, is employed to find optimal trajectories in various space missions. The overarching aim of this work is to obtain feasible global optimal solutions in terms of fuel consumption using a local optimization procedure. Consequently, the computational burden associated with the optimization process might be remarkably reduced. The solver requires a user-defined initial guess; thus a particular operational framework must be designed to obtain appropriate results, trying to discover the global minimum cost of the mission. The optimal solution found applying *fmincon* acutely depends on the provided initial guess, hence the proposed procedures must focus on building a favourable initial point. Obviously, that task becomes very complicated when working on complex problems like multiple Gravity Assist Maneuvers. Therefore, a different method should be modelled for each of the different considered functions: Direct transfer, DSMs, GAMs and DSMs + GAM.

The solution which is proposed in this work includes an Ant-Colony based optimization algorithm as part of the initial guess selection process. It is, in essence, a Metaheuristic-based method which imitates the ants' behaviour when searching for food.

The field of optimization of interplanetary trajectories has a rich literature. See the articles [17] and [6] for a general review. To cite a few works, in [11] the authors solve various missions that include, not only the direct transfer, but also GAMs and DSMs; furthermore, the MGA problem has been highly investigated, in references [12], [13], [14] and [16] some solved examples are presented.

1.3 Structure

Firstly, the elementary tools which are essential to start working on an interplanetary mission must be introduced. Lambert's Problem and the Ephemeris, as well as the orbital elements and the frame of reference, are explained in Section 2.
Secondly, in Section 3, how to design and to analyse an interplanetary trajectory is explained. The method of patched conics, Gravity assist Maneuvers and Deep Space Maneuvers are carefully submitted.
After that, the algorithms developed to solve every mission are presented in Section 4. The general procedure and the solver *fmincon* are introduced as well as the process to tackle every particular mission. Essentially the basis which is employed to build the fundamental Matlab scripts is explained.
Then, in Section 5, three strategies are proposed to overcome the lack of a global optimization solver. Basically, different procedures to feed *fmincon* with initial guesses are presented. Complexity grows from the first to the last algorithm, adding more sophisticated methods like the Ant-Colony based optimization algorithm. Summarising, the best algorithm provides *fmincon* with different and diverse initial guesses, then it focuses on the most auspicious ones and finally, it selects the solution associated with the lowest total impulse.
Eventually, in Section 6, every tool presented is applied to design several space missions. Three scenarios are considered, with the aim of reaching Mars, Saturn and Jupiter. Particular procedures are examined for each case, comparing them and selecting the most auspicious trajectories.

2 Basic Analysis Tools

The study of interplanetary missions is an extremely complex problem which involves sophisticated procedures. Accordingly, it is paramount to assure the correct definition of the basic tools that are required, in order to avoid major problems when working with more complicated methods. Firstly, it is necessary to define the unit of time employed, which is the Julian Day (Section 2.1), and to clarify the frame of reference used to express every variable of the problem, the Heliocentric frame of reference (Section 2.2). Once the importance of the orbital elements has been described in Section 2.2, it must be explained how to obtain them using the position and velocity vectors (Section 2.3) as well as how to work with the orbital elements in order to determine the position and velocity vectors (Section 2.4). Subsequently, the state vector of every planet under analysis must be identified for the time required, due to the fact that the orbits of the planets undergo slight changes over time. To tackle this problem, a method which obtains the Ephemeris of every planet at any given time is developed in Section 2.5. Finally, a procedure to solve Lambert's Problem is explained in Section 2.6, as a way of overcoming the need of calculating the transfer orbit between two desired points, having established the time of flight between them.

2.1 Julian Days

Julian Days were defined by Joseph Salinger in 1582, in order to avoid ambiguities when measuring time, such as leap years, allowing to unify the calendar. It consists on a count of days since an epoch fixed on 01/01/4713 BC at 12:00 UT. From the perspective of the Julian Date, days start at noon. Hours, minutes and seconds exceeding 12:00 UT are counted as fractions of a day.

It is easy to exactly obtain the Julian Date at 00:00 UT, having the year A, the month M and the day D, using equation 2.1, where $\lfloor x \rfloor$ means the integer part of x.

$$J_0 = 367A - \left\lfloor \frac{7A + 7\left\lfloor \frac{M+9}{12} \right\rfloor}{4} \right\rfloor + \left\lfloor \frac{275M}{9} \right\rfloor + D + 1721013.5. \tag{2.1}$$

If the date at any other UT is required, it can be calculated using:

$$JD = J_0 + \frac{UT}{24}. \tag{2.2}$$

For instance, it will be useful to consider the dates with respect to the J2000 epoch (01/01/2000 AC) which is the day 2451545.0.

2.2 Frame of reference

When working on an interplanetary mission, the Heliocentric frame of reference is used (figure 2.1). The location of the origin of this inertial reference system is coincident with the Sun and both, x-axis and y-axis, determine the Ecliptic Plane. The x-axis is defined as the alignment between the Earth and the Sun when the Vernal Equinox occurs. In this project, the bi-dimensional problem is considered, thus the analysis is carried

out as if every planetary orbit were on the Ecliptic Plane.

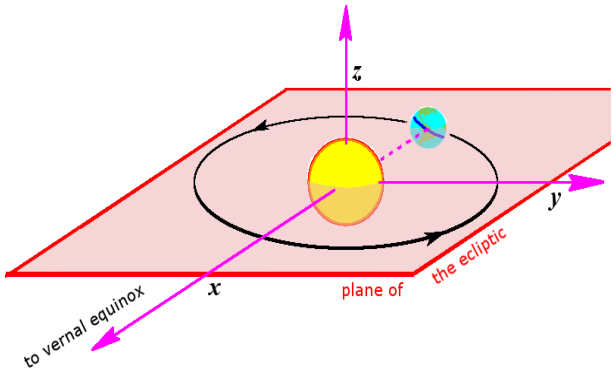

Figure 2.1 Heliocentric frame of reference.

Moreover, it is crucial to introduce the orbital elements, which are the only data we need, in addition to the epoch, to determine the position of a body at any given time. Two parameters are needed to define an orbit in the plane: eccentricity and angular momentum (the semimajor axis "a" is obtained from these two). Locating a point on the orbit requires a third parameter, the true anomaly, which supplies the elapsed time from the last periapsis flyby.
Then, three additional parameters are necessary to describe the orientation of an orbit in three dimensions. Firstly, the intersection of the orbital plane with the Equatorial Plane is located, calling that line the node line. The point on the line of nodes where the orbit passes above the Equatorial Plane from below it, is called the ascending node. Whereas the descending node is at the other end of the node line, where the orbit dives below the Equatorial Plane. The angle between the positive x-axis and the line of nodes is known as the right ascension of the ascending node, a positive number lying between 0° and 360°.
Subsequently, the inclination i is the dihedral angle between the orbital plane and the Equatorial Plane, measured according to the right-hand rule (counterclockwise around the line of nodes vector from the equator to the orbit). The inclination can also be obtained as the angle between the positive z-axis and the normal to the plane of the orbit. Since the specific angular momentum vector **h** is normal to the plane of the orbit, the inclination is the angle between the positive z-axis and **h**, and it is a positive number between 0° and 180°.
Eventually, it remains to locate the periapsis of the orbit, which lies at the intersection of the eccentricity vector **e** with the orbital path. The argument of periapsis ω is the angle between the line of nodes vector and the eccentricity vector, measured in the plane of the orbit, thus it is a positive number between 0° and 360°.

In brief, the Keplerian Orbital Elements are the following ones:

- a : Semi major axis [distance (AU)].
- e : Eccentricity [].
- θ : True anomaly [rad]. It is frequently replaced by the mean anomaly M.
- ω : Argument of periapsis [rad].
- Ω : Right ascension of the ascending node [rad].
- i : Inclination [rad].

However, in certain special cases the Classical Orbital Elements are not properly defined, such as circular and equatorial orbits, hence they may be replaced with different elements such as the longitude of periapsis ($\bar{\omega} = \Omega + \omega$), the argument of latitude (u $= \omega + \theta$) and the true longitude ($\lambda_T = \omega + \theta + \Omega$).

2.3 Orbital elements determination from position and velocity vectors

Having the position and velocity vectors expressed on the Heliocentric reference system (**r** and **v**), it is possible to obtain the orbital elements of the orbit under study. First, the specific energy, the specific angular momentum and the eccentricity vectors are calculated:

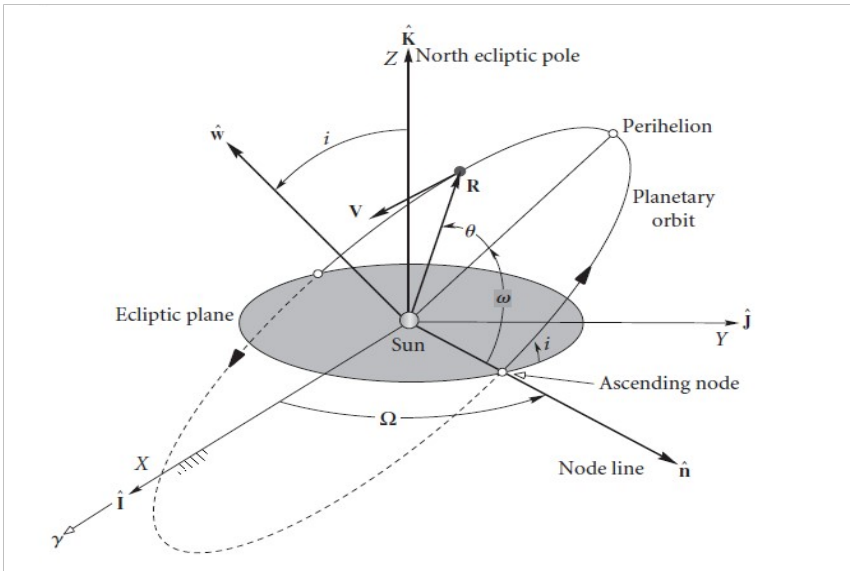

Figure 2.2 Orbital elements in the Heliocentric frame of reference, extracted from [2].

$$\varepsilon = \frac{v^2}{2} - \frac{\mu}{r}, \tag{2.3}$$

$$\mathbf{h} = \mathbf{r} \times \mathbf{v}, \tag{2.4}$$

$$\mathbf{e} = \frac{\mathbf{r} \times \mathbf{v}}{\mu} - \frac{\mathbf{r}}{r}, \tag{2.5}$$

where r and v are the norm of the vectors $\mathbf{r}$ and $\mathbf{v}$, respectively, and μ is the gravitational parameter. For instance, in order to analyse heliocentric orbits, the gravitational parameter associated with the Sun is used. It is defined as $\mu_\odot = GM_\odot$, being $M_\odot$ the mass of the Sun. Therefore, the semimajor axis and the eccentricity of the orbit are obtained.

$$a = -\frac{\mu}{2\varepsilon}, \qquad e = |\mathbf{e}|. \tag{2.6}$$

At this point, the line of nodes, which is the intersection of the orbital plane with the Equatorial Plane, can be determined as:

$$\mathbf{n} = \frac{\mathbf{k} \times \mathbf{h}}{|\mathbf{h}|}. \tag{2.7}$$

Depending on the case under analysis, it might be auspicious to calculate different orbital elements. Since planetary orbits around the Sun are considered elliptical trajectories, it must be borne in mind that, under Keplerian assumptions, circular orbits (e=0) are unlikely to happen.

- General case:

$$\theta = \cos^{-1}\left(\frac{\mathbf{r}\cdot\mathbf{e}}{r\,e}\right), \quad \text{if } \mathbf{r}\cdot\mathbf{v} \geq 0. \qquad \theta = 2\pi - \cos^{-1}\left(\frac{\mathbf{r}\cdot\mathbf{e}}{r\,e}\right), \quad \text{if } \mathbf{r}\cdot\mathbf{v} < 0. \tag{2.8}$$

$$\Omega = \cos^{-1}(n_x), \quad \text{if } n_y \geq 0. \qquad \Omega = 2\pi - \cos^{-1}(n_x), \quad \text{if } n_y < 0. \tag{2.9}$$

$$\omega = \cos^{-1}\left(\frac{\mathbf{n}\cdot\mathbf{e}}{e}\right), \quad \text{if } e_z \geq 0. \qquad \omega = 2\pi - \cos^{-1}\left(\frac{\mathbf{n}\cdot\mathbf{e}}{e}\right), \quad \text{if } e_z < 0. \tag{2.10}$$

- Case 1 (i = 0°):

$$\theta = \cos^{-1}\left(\frac{\mathbf{r}\cdot\mathbf{e}}{\mathrm{r\,e}}\right), \quad \text{if } \mathbf{r}\cdot\mathbf{v} \geq 0. \qquad \theta = 2\pi - \cos^{-1}\left(\frac{\mathbf{r}\cdot\mathbf{e}}{\mathrm{r\,e}}\right), \quad \text{if } \mathbf{r}\cdot\mathbf{v} < 0. \tag{2.11}$$

$$\bar{\omega} = \cos^{-1}\left(\frac{\mathrm{e_x}}{\mathrm{e}}\right), \quad \text{if } \mathrm{e_y} \geq 0. \qquad \bar{\omega} = 2\pi - \cos^{-1}\left(\frac{\mathrm{e_x}}{\mathrm{e}}\right), \quad \text{if } \mathrm{e_y} < 0. \tag{2.12}$$

- Case 2 (e = 0):

$$\Omega = \cos^{-1}(\mathrm{n_x}), \quad \text{if } \mathrm{n_y} \geq 0. \qquad \Omega = 2\pi - \cos^{-1}(\mathrm{n_x}), \quad \text{if } \mathrm{n_y} < 0. \tag{2.13}$$

$$u = \cos^{-1}\left(\frac{\mathbf{r}\cdot\mathbf{n}}{\mathrm{r\,n}}\right), \quad \text{if } \mathrm{r_z} \geq 0. \qquad u = 2\pi - \cos^{-1}\left(\frac{\mathbf{r}\cdot\mathbf{n}}{\mathrm{r\,n}}\right), \quad \text{if } \mathrm{r_z} < 0. \tag{2.14}$$

- Case 3 (i = 0°, e = 0):

$$\lambda_{\mathrm{T}} = \cos^{-1}\left(\frac{\mathrm{r_x}}{\mathrm{r}}\right), \quad \text{if } \mathrm{r_y} \geq 0. \qquad \lambda_{\mathrm{T}} = 2\pi - \cos^{-1}\left(\frac{\mathrm{r_x}}{\mathrm{r}}\right), \quad \text{if } \mathrm{r_y} < 0. \tag{2.15}$$

2.4 Position and velocity vectors determination from orbital elements

The aim of this section is to calculate the position and velocity vectors at a desired time, given the orbital elements at that time. First, both vectors are obtained on the Perifocal reference system and, after that, they must be transformed to the Heliocentric system. Consequently, in order to express those vectors on the Heliocentric frame, a rotation matrix is needed. Vectors expressed on the Perifocal reference system are denoted using the superscript "P". Similarly, the superscript "H" is used when working with vectors in the Heliocentric frame.

$$\mathbf{r}^H = \left[\mathbf{C}_H^F\right]^T \mathbf{r}^F, \tag{2.16}$$

$$\mathbf{v}^H = \left[\mathbf{C}_H^F\right]^T \mathbf{v}^F. \tag{2.17}$$

Depending on the particular case under analysis, the rotation matrix as well as the position and velocity vectors in the Perifocal frame, may be modified.

- General case:

$$\mathbf{r}^F = \left(\frac{\mathrm{p}}{1+\mathrm{e}\,\cos(\theta)}\right)\begin{bmatrix}\cos(\theta)\\ \sin(\theta)\\ 0\end{bmatrix}, \tag{2.18}$$

$$\mathbf{v}^F = \sqrt{\frac{\mu}{\mathrm{p}}}\begin{bmatrix}-\sin(\theta)\\ \mathrm{e}+\cos(\theta)\\ 0\end{bmatrix}, \tag{2.19}$$

$$\mathbf{C}_H^F = \begin{bmatrix}\cos(\Omega)\cos(\omega)-\sin(\Omega)\sin(\omega)\cos(\mathrm{i}) & \sin(\Omega)\cos(\omega)-\cos(\Omega)\sin(\omega)\cos(\mathrm{i}) & \sin(\omega)\sin(\mathrm{i})\\ -\cos(\Omega)\sin(\omega)-\sin(\Omega)\cos(\omega)\cos(\mathrm{i}) & -\sin(\Omega)\sin(\omega)+\cos(\Omega)\cos(\omega)\cos(\mathrm{i}) & \cos(\omega)\sin(\mathrm{i})\\ \sin(\Omega)\sin(\omega) & -\cos(\Omega)\sin(\mathrm{i}) & \cos(\mathrm{i})\end{bmatrix}. \tag{2.20}$$

- Case 1 (i = 0°): The position and velocity vectors in the Perifocal system are the ones shown on equations 2.18 and 2.19.

$$\mathbf{C}_H^F = \begin{bmatrix}\cos(\bar{\omega}) & \sin(\bar{\omega}) & 0\\ -\sin(\bar{\omega}) & \cos(\bar{\omega}) & 0\\ 0 & 0 & 1\end{bmatrix}. \tag{2.21}$$

- Case 2 (e = 0):

$$\mathbf{r}^F = a \begin{bmatrix} \cos(\mathrm{u}) \\ \sin(\mathrm{u}) \\ 0 \end{bmatrix}, \tag{2.22}$$

$$\mathbf{v}^F = \sqrt{\frac{\mu}{\mathrm{a}}} \begin{bmatrix} -\sin(\mathrm{u}) \\ \cos(\mathrm{u}) \\ 0 \end{bmatrix}, \tag{2.23}$$

$$\mathbf{C}_H^F = \begin{bmatrix} \cos(\Omega)\cos(\mathrm{u}) - \sin(\Omega)\sin(\mathrm{u})\cos(\mathrm{i}) & \sin(\Omega)\cos(\mathrm{u}) - \cos(\Omega)\sin(\mathrm{u})\cos(\mathrm{i}) & \sin(\mathrm{u})\sin(\mathrm{i}) \\ -\cos(\Omega)\sin(\mathrm{u}) - \sin(\Omega)\cos(\mathrm{u})\cos(\mathrm{i}) & -\sin(\Omega)\sin(\mathrm{u}) + \cos(\Omega)\cos(\mathrm{u})\cos(\mathrm{i}) & \cos(\mathrm{u})\sin(\mathrm{i}) \\ \sin(\Omega)\sin(\mathrm{u}) & -\cos(\Omega)\sin(\mathrm{i}) & \cos(\mathrm{i}) \end{bmatrix}. \tag{2.24}$$

- Case 3 (i= 0°, e = 0):

$$\mathbf{r}^F = a \begin{bmatrix} \cos(\lambda_\mathrm{T}) \\ \sin(\lambda_\mathrm{T}) \\ 0 \end{bmatrix}, \tag{2.25}$$

$$\mathbf{v}^F = \sqrt{\frac{\mu}{\mathrm{a}}} \begin{bmatrix} -\sin(\lambda_\mathrm{T}) \\ \cos(\lambda_\mathrm{T}) \\ 0 \end{bmatrix}, \tag{2.26}$$

$$\mathbf{C}_H^F = \begin{bmatrix} \cos(\lambda_\mathrm{T}) & \sin(\lambda_\mathrm{T}) & 0 \\ -\sin(\lambda_\mathrm{T}) & \cos(\lambda_\mathrm{T}) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.27}$$

2.5 Planetary Ephemeris

For the purpose of designing realistic interplanetary missions, it is necessary to determine the state vector of a planet at any given time. Using the method developed in reference [7], it is possible to obtain the orbital elements of the planets employing their rates of change per century with respect to the J2000 epoch (1 January 2000, 12:00 UT). The foremost aspect of this procedure, which must be taken into account, is the fact that the valid time interval goes from 1800 AC to 2050 AC. Then, having the orbital elements, the state vector can be inferred as it has been shown.

The method consists on a linear model, where the orbital elements are calculated as:

$$(\mathrm{a,e,i,}\Omega,\bar{\omega},\mathrm{L}) = (\mathrm{a_0,e_0,i_0,}\Omega_0,\bar{\omega}_0,L_0) + \frac{\partial}{\partial t}(\mathrm{a,e,i,}\Omega,\bar{\omega},\mathrm{L}) \cdot \mathrm{T}, \tag{2.28}$$

where $\mathrm{L} = \bar{\omega} + \mathrm{M}$, is the mean longitude and T_0 is the number of centuries past J2000, obtained using the desired Julian day (JD) as:

$$\mathrm{T} = \frac{\mathrm{JD} - 2451545}{36525}. \tag{2.29}$$

Once the mean anomaly M has been calculated using L and $\bar{\omega}$, the eccentric anomaly E is determined via Kepler's Equation (2.30).

$$\mathrm{M} = \mathrm{E} - \mathrm{e}\sin(\mathrm{E}). \tag{2.30}$$

Finally, the true anomaly θ is obtained from equation 2.31.

$$\tan(\theta/2) = \sqrt{\frac{1+\mathrm{e}}{1-\mathrm{e}}}\tan(\mathrm{E}/2). \tag{2.31}$$

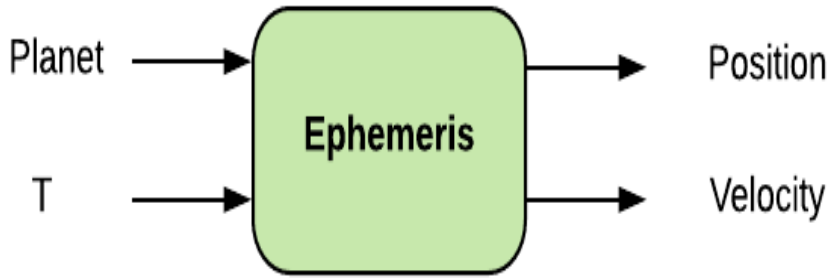

Figure 2.3 Schematic of the Ephemeris tool.

2.6 Lambert's Problem

The appropriate analysis of an interplanetary mission leads to a comparison of myriad different transfer orbits. A way of tackling this question is solving Lambert's Problem, which consists on determining a transfer orbit from two position vectors and a desired time of flight between them. Having said that, a tool which solves Lambert's Problem becomes essential for the aim of this project.

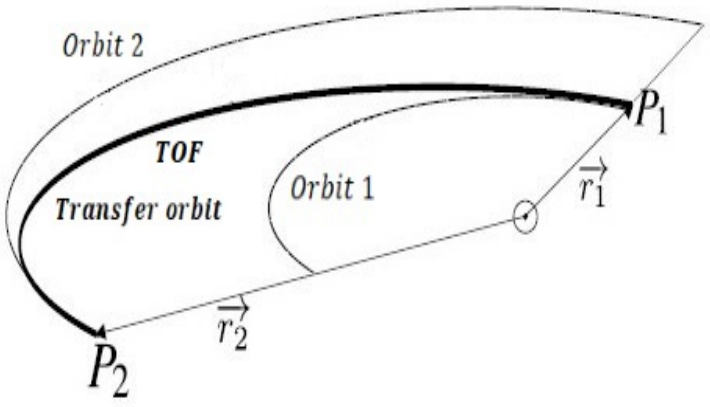

Figure 2.4 Example of a solution to Lambert's Problem.

Nowadays, solutions to Lambert's Problem are used extensively in the orbital targeting problem, specifically in areas such as preliminary orbit determination and rendezvous analysis. Due to the growing complexity of the space missions, it is necessary to determine an efficient and robust Lambert solution algorithm. A large number of approaches have been formulated to solve this problem, each one involving a particular procedure and having a different associated efficiency. Some of them are the Universal Variable method, Battin's formulation or Sun's method.

Once having investigated and analysed some of the possible solutions using reference [19], the method proposed in reference [2] is selected to solve Lambert's Problem in this project, due to its simplicity despite the fact that an initial guess is needed.

Firstly, given the position vectors $\mathbf{r}_1$ and $\mathbf{r}_2$ of the points that must be linked using a transfer orbit (P_1 and P_2 as illustrated in figure 2.4), $\mathbf{r}_1$ and $\mathbf{r}_2$ determine the change in the true anomaly $\Delta\theta$, since:

$$\cos(\Delta\theta) = \frac{\mathbf{r}_1 \cdot \mathbf{r}_2}{r_1\, r_2}, \tag{2.32}$$

where

$$r_1 = \sqrt{\mathbf{r}_1 \cdot \mathbf{r}_1}, \qquad r_2 = \sqrt{\mathbf{r}_2 \cdot \mathbf{r}_2}. \tag{2.33}$$

Another parameter α, which dictates the transfer angle to be less or greater than 180° is determined from the third component of the specific angular momentum of the transfer orbit. That is:

$$\alpha = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \cdot (\mathbf{r}_1 \times \mathbf{r}_2). \tag{2.34}$$

Moreover, it would be possible to select a direct orbit ($i \leq 90^\circ$) or a retrograde one ($i > 90^\circ$). Nonetheless, only direct orbits are considered.

To start the procedure's explanation, it must be clarified that using the Lagrange coefficients f and g, if the position and velocity of an orbiting body are known at a given instant, it is possible to obtain them at any later time in terms of the initial values. Consequently:

$$\mathbf{r}_2 = f\,\mathbf{r}_1 + g\,\mathbf{v}_1, \tag{2.35}$$
$$\mathbf{v}_2 = \dot{f}\,\mathbf{r}_1 + \dot{g}\,\mathbf{v}_1. \tag{2.36}$$

Solving these equations and considering that $f\dot{g} - \dot{f}g = 1$, both velocities are obtained in terms of both positions and the Lagrange coefficients:

$$\mathbf{v}_1 = \frac{1}{g}(\mathbf{r}_2 - f\,\mathbf{r}_1) \tag{2.37}$$
$$\mathbf{v}_2 = \frac{1}{g}(\dot{g}\,\mathbf{r}_2 - \mathbf{r}_1) \tag{2.38}$$

Therefore, Lambert's Problem is solved once the Lagrange coefficients f, g and $\dot{g}$ are determined. The Lagrange coefficients f and g, as well as their time derivatives are listed as functions of the change in the true anomaly:

$$f = 1 - \frac{\mu r_2}{h^2}(1-\cos(\Delta\theta)), \qquad g = \frac{r_1 r_2}{h}\sin(\Delta\theta), \tag{2.39}$$
$$\dot{f} = \frac{\mu}{h}\frac{1-\cos(\Delta\theta)}{\sin(\Delta\theta)}\left[\frac{\mu}{h^2}(1-\cos(\Delta\theta)) - \frac{1}{r_1} - \frac{1}{r_2}\right], \qquad \dot{g} = 1 - \frac{\mu r_1}{h^2}(1-\cos(\Delta\theta)). \tag{2.40}$$

It is also possible to express these quantities in terms of the universal anomaly χ:

$$f = 1 - \frac{\chi^2}{r_1}C(z), \qquad g = \Delta t - \frac{a}{\sqrt{\mu}}\chi^3 S(z), \tag{2.41}$$
$$\dot{f} = \frac{\sqrt{\mu}}{r_1 r_2}\chi\,[zS(z)-1], \qquad \dot{g} = 1 - \frac{\chi^2}{r_2}C(z), \tag{2.42}$$

where $z = \beta\chi^2$ (β is the reciprocal of the semimajor axis of the unknown orbit). A relationship between Δt and $\Delta\theta$ can be found by equating the two expressions for g:

$$\frac{r_1 r_2}{h}\sin(\Delta\theta) = \Delta t - \frac{1}{\sqrt{\mu}}\chi^3 S(z). \tag{2.43}$$

Then, to eliminate the angular momentum h, both expressions for f are equated, obtaining:

$$h = \sqrt{\frac{\mu r_1 r_2 (1-\cos(\Delta\theta))}{\chi^2 C(z)}} \tag{2.44}$$

Accordingly, equation 2.43 is simplified :

$$\sqrt{\mu}\Delta t = \chi^3 S(z) + A\chi\sqrt{C(z)}, \tag{2.45}$$

where the symbol A refers to:

$$A = \sin(\Delta\theta)\sqrt{\frac{r_1 r_2}{1-\cos(\Delta\theta)}}. \tag{2.46}$$

In order to discover a relationship between z and χ, the function y(z) is defined as:

$$y(z) = r_1 + r_2 + A\frac{zS(z)-1}{\sqrt{C(z)}} \tag{2.47}$$

Hence, $\chi = \sqrt{y(z)/C(z)}$. Substituting it back into equation 2.48 yields:

$$\sqrt{\mu}\Delta t = \left[\frac{y(z)}{C(z)}\right]^{\frac{3}{2}} S(z) + A\sqrt{y(z)}, \tag{2.48}$$

After this, it is possible to iteratively apply Newton's method to solve for z, using the function F(z):

$$F(z) = \left[\frac{y(z)}{C(z)}\right]^{\frac{3}{2}} S(z) + A\sqrt{y(z)} - \sqrt{\mu}\Delta t. \tag{2.49}$$

Its derivative is defined as:

$$\frac{\partial F(z)}{\partial z} = \begin{cases} \left[\frac{y(z)}{C(z)}\right]^{\frac{3}{2}} \left[\frac{1}{2z}\left(C(z) - \frac{3S(z)}{2C(z)}\right) + \frac{3S(z)^2}{4C(z)}\right] + \frac{A}{8}\left[3\frac{S(z)}{C(z)}\sqrt{y(z)} + A\sqrt{\frac{C(z)}{y(z)}}\right], & \text{if } z \neq 0. \\ \frac{\sqrt{2}}{40}y(0)^{\frac{3}{2}} + \frac{A}{8}\left[\sqrt{y(0)} + A\sqrt{\frac{1}{2y(0)}}\right], & \text{if } z = 0. \end{cases} \tag{2.50}$$

Lastly, the functions C and S are presented:

$$C(z) = \begin{cases} \frac{1-\cos(\sqrt{z})}{z}, & \text{if } z > 0. \\ \frac{\cosh(\sqrt{-z})-1}{-z}, & \text{if } z < 0. \\ \frac{1}{2}, & \text{if } z = 0. \end{cases} \tag{2.51}$$

$$S(z) = \begin{cases} \frac{\sqrt{z}-\sin(\sqrt{z})}{(\sqrt{z})^3}, & \text{if } z > 0. \\ \frac{\sinh(\sqrt{-z})-(\sqrt{-z})}{(\sqrt{-z})^3}, & \text{if } z < 0. \\ \frac{1}{6}, & \text{if } z = 0. \end{cases} \tag{2.52}$$

At this point, an initial guess is required to start iterating with the optimisation parameter of the algorithm (z). In order to improve the efficiency of the algorithm, a process to find a favourable initial guess is included. It consists on, starting from a known solution (it is known that z is positive for elliptic orbits and negative for hyperbolic orbits), increasing its value with a non accurate step, that is to say not as small as the optimisation one, until the objective function changes its sign. Hence the optimal solution would be close to that initial guess.

Having calculated an initial guess, the iteration procedure begins. The iteration ends when a solution is found, having established a termination tolerance. In addition to that tolerance, a maximum number of

iterations is set. Therefore, after having used Lambert's Algorithm, it is necessary to check whether the solution is correct or the algorithm has stopped by reaching the maximum number of iterations fixed, thus the solution could not be used.

Summarising, this algorithm calculates the velocities associated with both points, P_1 and P_2, in the transfer orbit. Eventually, using the transformations explained in Section 2.3, the orbital elements are determined.

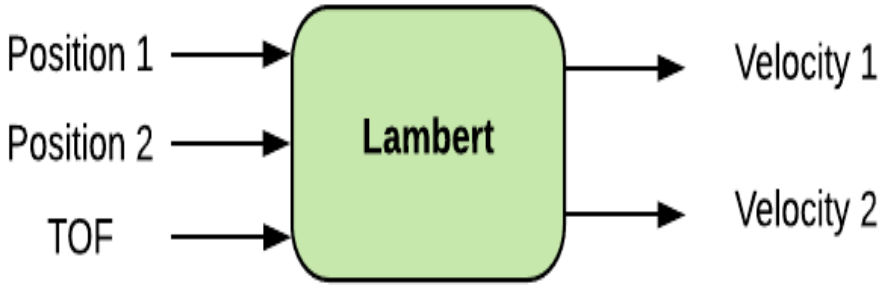

Figure 2.5 Schematic of the Lambert tool.

3 Interplanetary Trajectories

The overall target of this project is to optimize interplanetary trajectories, hence the procedure which has been followed to calculate the required orbits must be explained. The method of patched conics allows to solve a problem which involves more than two bodies. In order to apply this method, developed in Section 3.2, it is necessary to introduce a meaningful concept, the Sphere of Influence, in Section 3.1.

Moreover, since the work is carried out on the 2-D problem, planetary orbits around the Sun are considered to be on the Ecliptic Plane. Therefore, only two vector components for each variable have to be accounted for.

To conduct the analysis of planetary missions, general data of every planet is required. This general data is summarised on table 3.1, where R_P is the planet's radius, L_P is the mean distance from the Sun to the planet and μ_P is the planet's gravitational parameter.

Planet	$\mathbf{R_P}$ (km)	$\mathbf{L_P}$ (AU)	$\mu_\mathbf{P}$ $\left(\mathrm{km}^3/\mathrm{s}^2\right)$
Mercury	2439.7	0.3871	22032.1
Venus	6051.8	0.7233	324858.8
Earth	6378.14	1	398600.4
Mars	3397	1.5237	42828.3
Jupiter	71492	5.2033	126711995.4
Saturn	60268	9.5808	37939519.7
Uranus	25559	19.2709	5780158.5
Neptune	24764	30.1927	6871307.8

Table 3.1 Planetary general data.

3.1 Sphere of influence

In accordance with reference [18], the Sphere of Influence is the region in which the gravitational influence of a body prevails over the influence of other bodies. Similarly, the Sphere of Influence radius is a reasonable estimate of the distance beyond which the Sun's gravitational attraction dominates that of a planet. In order to appraise that radius, the Three-Body Problem is considered. It comprises a planet "p" of mass m_p, the Sun "s" of mass m_s and a space vehicle "v" of mass m_v (represented on figure 3.1). To give an illustration, when the vehicle is close to the Sun, the effect of the planet can be treated as a perturbation on the Keplerian motion of the vehicle around the Sun:

$$\ddot{\mathbf{R}}_v = -\mu_s \frac{\mathbf{R}_v}{\mathbf{R}_v^3} + \mu_p \left| \frac{\mathbf{r}}{r^3} - \frac{\mathbf{R}}{R^3} \right| = \mathbf{g}_s + \mathbf{g}'_p, \tag{3.1}$$

where r, R and R_v are the norm of the vectors **r**, **R** and $\mathbf{R}_v$, respectively. Likewise, if the vehicle is close to the planet, the effect of the Sun can be modeled as a perturbation:

$$\ddot{\mathbf{r}} = -\mu_p \frac{\mathbf{r}}{r^3} + \mu_s \left| \frac{\mathbf{R}_v}{R_v^3} - \frac{\mathbf{R}}{R^3} \right| = \mathbf{g}_p + \mathbf{g}_s'. \tag{3.2}$$

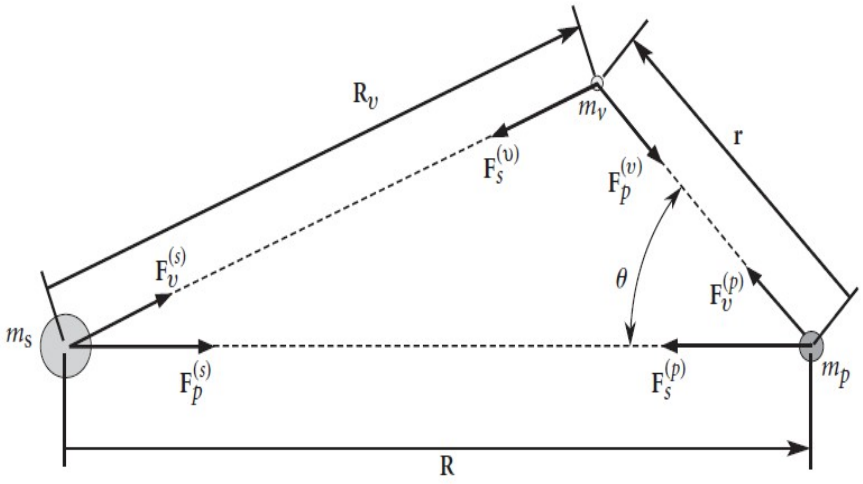

Figure 3.1 Three-Body Problem representation, pulled from [2].

As a result, the geometric place of the points where $\frac{g_p'}{g_s} = \frac{g_s'}{g_p}$ separates the influence zones of the Sun and the planet. Nevertheless, the Sun is the dominant celestial body in the Solar System, over 1000 times more massive than Jupiter and over 300000 times more massive than Earth. Therefore, as $m_s >> m_p$, that geometric place is similar to a sphere around the planet, which is called Sphere of Influence.

Inside the Sphere of Influence, it is possible to consider that the only body which affects the vehicle is the planet. Consequently, the vehicle travels an unperturbed Keplerian path around the planet, simplifying the problem to a Two-Body Problem. On the other hand, outside the Sphere, the only influence is the Sun, thus it is assumed that the spacecraft follows an unperturbed Keplerian orbit around the Sun. The radius of the Sphere is given by:

$$R_{e_p} = L_p \left(\frac{\mu_p}{\mu_s} \right)^{\frac{2}{5}}. \tag{3.3}$$

For every planet, it is true that $R_p << R_{e_p} << L_p$, hence it is possible to simplify the interplanetary problem. When working on the planet-centric phase, $R_{e_p} \to \infty$, and from the heliocentric point of view, $R_{e_p} \to 0$, so the Sphere of Influence is considered as a point (planet position).

3.2 Method of patched conics

The method of patched conics is employed to divide the mission into three parts: the hyperbolic departure trajectory relative to the home planet, the cruise ellipse relative to the Sun and the hyperbolic arrival trajectory relative to the target planet. In order to illustrate the method, a mission from the Earth to Planet 2 is analysed.

3.2.1 Planetary departure, Geocentric phase

The vehicle departs from a park orbit at an altitude h_{park1} from the Earth's surface ($r_{park1} = R_\oplus + h_{park1}$). At first, it must escape the gravitational pull of the Earth as it is inside its Sphere of Influence, travelling a hyperbolic trajectory whose focus is the Earth, in a Geocentric reference system. ΔV_{ini} is the impulse required at the park orbit to insert the vehicle in a scape trajectory, $V_\oplus$ is the Earth's velocity and $V_{\infty 1}$ is the

hyperbolic excess speed (when the spacecraft arrives at the limit of the Earth's Sphere of Influence).

$$\Delta V_{ini} = \sqrt{\frac{2\mu_{\oplus}}{r_{park1}} + (V_{\infty 1})^2} - \sqrt{\frac{\mu_{\oplus}}{r_{park1}}}. \tag{3.4}$$

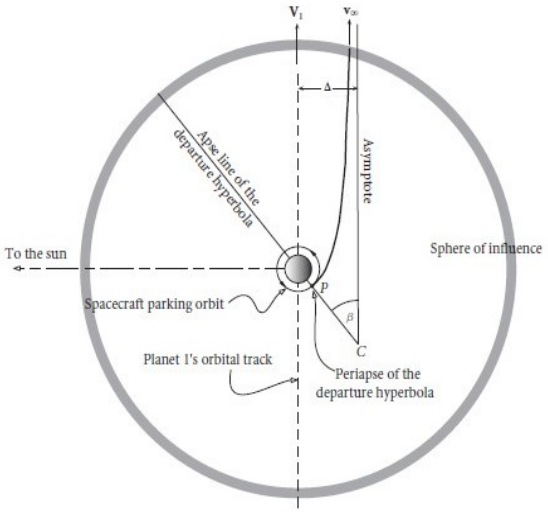

Figure 3.2 Departure of a spacecraft on a mission from an inner planet to an outer planet, extracted from [2].

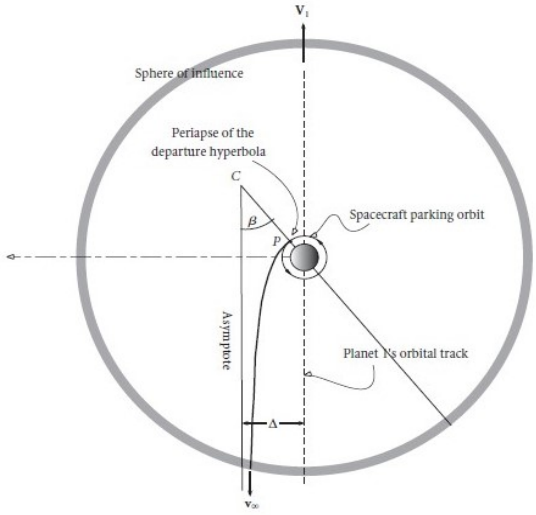

Figure 3.3 Departure of a spacecraft on a mission from an outer planet to an inner planet, extracted from [2].

3.2.2 Heliocentric phase

The Heliocentric phase consists on a trajectory, with the Sun as the focus, which starts when the spacecraft leaves the Earth's Sphere of Influence and ends when it arrives at the Sphere of Influence of Planet 2. After escaping from the Earth's gravitational pull, in the Heliocentric reference system, the vehicle travels with a velocity $V_0^H = V_{\oplus} + V_{\infty 1}$ if it goes to a superior planet and $V_0^H = V_{\oplus} - V_{\infty 1}$ if it travels to an inferior planet. Furthermore, it arrives at the Sphere of Influence of Planet 2 having a velocity V_1^H. The semimajor axis of the heliocentric conic is a, and the symbol $\odot$ refers to the Sun.

$$V_{\infty 1} = \pm \left(\sqrt{\frac{2\mu_{\odot}}{L_{\oplus}} - \frac{\mu_{\odot}}{a}} - \sqrt{\frac{\mu_{\odot}}{L_{\oplus}}} \right) = \pm \left(V_0^H - V_{\oplus} \right). \tag{3.5}$$

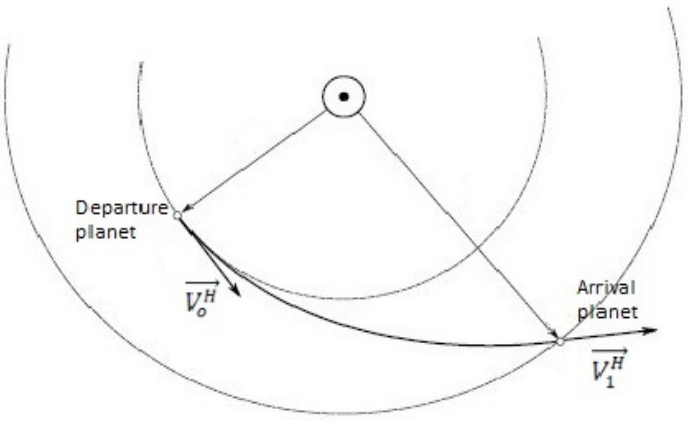

Figure 3.4 Heliocentric phase representation.

3.2.3 Planetary arrival phase

Finally, the space vehicle goes into the Sphere of Influence of Planet 2. Then, it travels a hyperbolic trajectory, whose focus is Planet 2, until it gets to the desired altitude from the surface of that planet, h_{park2}. The hyperbolic excess speed associated with this phase is $V_{\infty 2}$, V_{P2} is the velocity of Planet 2 and ΔV_{end} is the impulse required to insert the spacecraft at the desired parking orbit.

$$V_{\infty 2} = \pm \left(V_1^H - V_{P2} \right). \tag{3.6}$$

$$\Delta V_{end} = \sqrt{\frac{2\mu_{P2}}{r_{park2}} + (V_{\infty 2})^2} - \sqrt{\frac{\mu_{P2}}{r_{park2}}}. \tag{3.7}$$

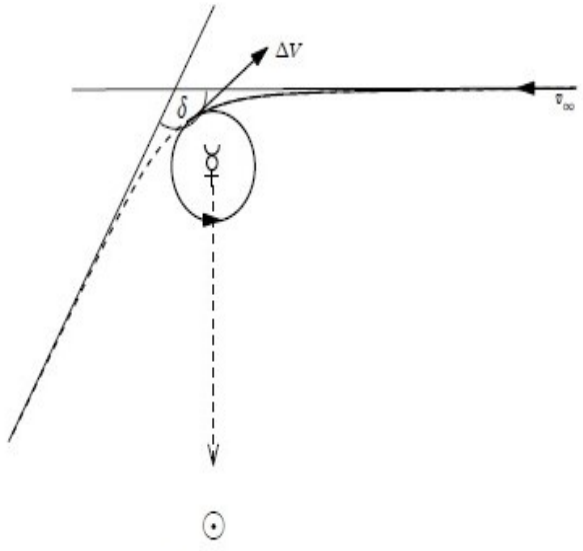

Figure 3.5 Arrival of a spacecraft on a mission from an outer planet to Mercury, extracted from [8].

3.3 Hohmann Transfer

The Hohmann transfer is a maneuver which connects two circular orbits. Although it is the simplest and the cheapest two-impulse maneuver, it is only applicable if both orbits are coplanar. Nevertheless, the problem has been simplified to a 2-D one. Provided that, planetary orbits are considered elliptical and coplanar, hence this maneuver may become useful.

Using a Hohmann transfer ellipse is the most efficient way, in terms of fuel consumption, for a spacecraft to travel from one planetary orbit to another. The departure point is at the periapsis of the transfer ellipse and the arrival point is at the apoapsis. The Hohmann transfer orbit is a semi-ellipse, consequently it is actually easy to obtain the semimajor axis, a_H, and the time of flight of the maneuver, TOF_H, which is given by the semi-period associated with the ellipse.

$$a_H = \frac{R_1 + R_2}{2}, \qquad TOF_H = \pi \sqrt{\frac{a_H^3}{\mu}}, \tag{3.8}$$

where R_1 and R_2 and the radius of both circular orbits involved, as it is illustrated in figure 3.6, and μ is the gravitational parameter. As it has been said, when working with elliptical orbits it is not completely correct to talk about Hohmann's transfers. Nonetheless, the Hohmann Transfer can be adapted to elliptical orbits and it is used to compare the orbits obtained later in chapter 6.

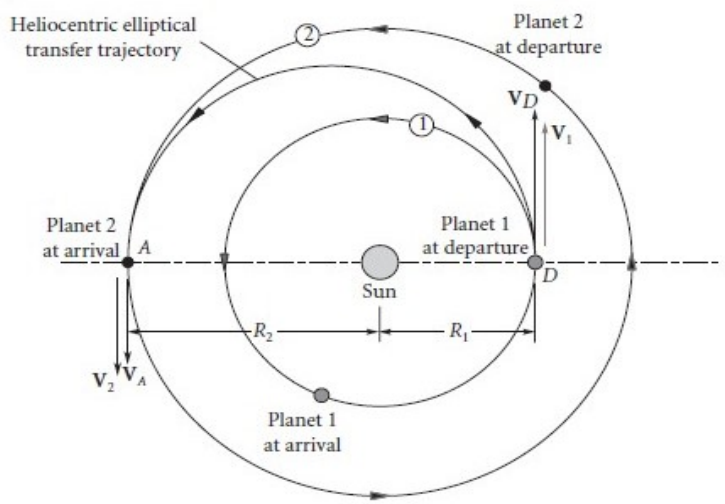

Figure 3.6 Hohmann Transfer example, pulled from [2].

3.4 Gravity Assist Maneuver (GAM)

There are many cases, when colossal distances separate both the initial and the final planet, where a simple two-impulse maneuver becomes unfeasible in terms of fuel expense. Particularly, there may not exist a space launcher which provides the energy needed for that mission or the cost of the mission might be unaffordable.

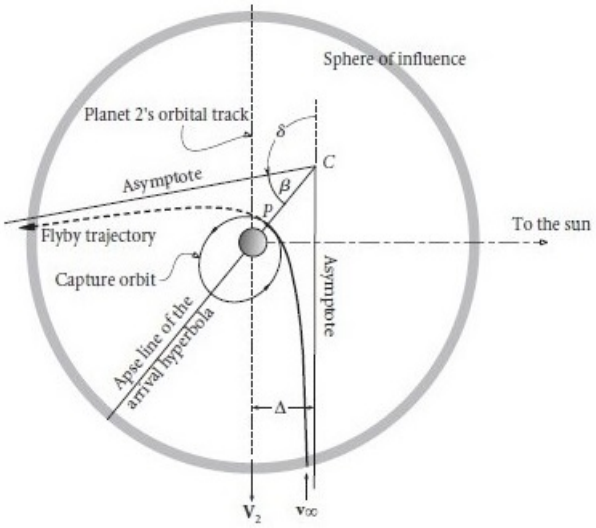

Figure 3.7 Flyby from an inner planet to an outer planet, extracted from [2].

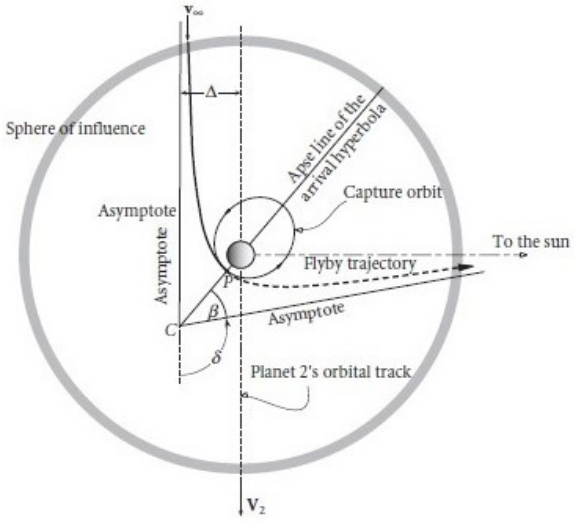

Figure 3.8 Flyby from an outer planet to an inner planet, extracted from [2].

Nevertheless, several spacecrafts have travelled large distances in space. In particular, the Voyager II escaped the Solar System, avoiding the limitations mentioned. This is possible thanks to the use of Gravity Assist Maneuvers, which consist on gaining speed with a flyby over an intermediate planet. Fundamentally, the flyby modifies, not only the norm, but also the direction of the heliocentric velocity of a space vehicle. This maneuver changes the kinetic energy of the vehicle without using propellant, thus it does not entail an extra cost.

Before heading to the procedure's explanation, the nomenclature used in this section must be detailed.

- $\mathbf{V}_1^{\mathrm{H}}$: Heliocentric velocity vector of the vehicle approaching the planet of the flyby before the maneuver.

- $\mathbf{V}_2^H$: Heliocentric velocity vector of the vehicle leaving the planet of the flyby after the maneuver.
- $\mathbf{V}_1^P$: Initial velocity vector relative to the planet of the flyby. It is the excess velocity of the initial hyperbolic orbit.
- $\mathbf{V}_2^P$: Final velocity vector relative to the planet of the flyby. It is the excess velocity of the final hyperbolic orbit.
- $\mathbf{V}_P$: Heliocentric velocity vector of the planet "P".
- r_p: periapse radius of the hyperbola.
- μ_P: gravitational parameter of the planet "P".
- δ: turn angle of the asymptote of the hyperbola.
- Generally V is used to name the norm of the vector $\mathbf{V}$.

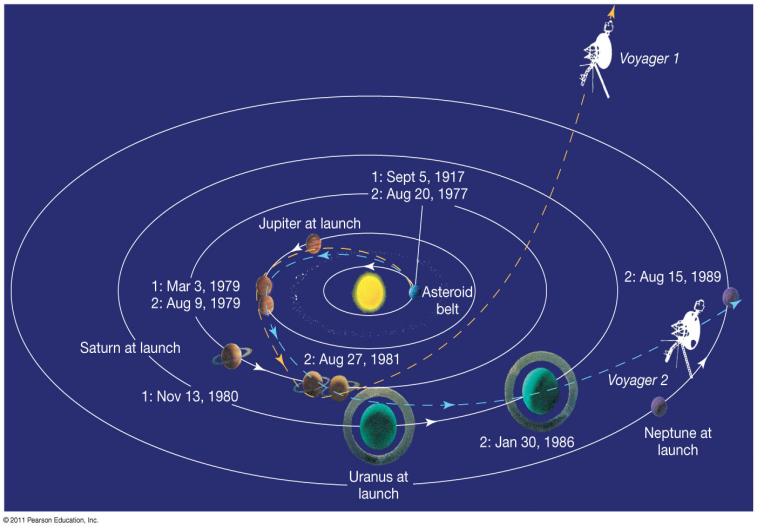

Figure 3.9 Voyager I and II mission overview, pulled from [1].

As previously noted, when the spacecraft enters the Sphere of Influence of the planet where the flyby takes place, it describes a hyperbolic orbit whose excess velocity is $\mathbf{V}_1^P$. In this maneuver, the norm of the excess velocity vector stays constant, thus $V_\infty = V_1^P = V_2^P$.

Firstly, it is easy to obtain $\mathbf{V}_1^P$ and $\mathbf{V}_1^P$, as it is shown in figure 3.10:

$$\mathbf{V}_1^P = \mathbf{V}_1^H - \mathbf{V}_P. \tag{3.9}$$
$$\mathbf{V}_2^P = \mathbf{V}_2^H - \mathbf{V}_P. \tag{3.10}$$

Secondly, the impulse obtained from the Gravity Assist Maneuver can be calculated as:

$$\Delta V = \frac{2V_\infty}{1 + \frac{r_p V_\infty^2}{\mu_P}} = 2V_\infty \sin(\delta/2). \tag{3.11}$$

The turn angle δ can also be calculated from the eccentricity of the hyperbolic orbit as $\delta = 2\arcsin(1/e)$. On top of that, the value of this angle depends on whether the vehicle is going to a superior or to an inferior planet.

In brief, the impulse obtained from the maneuver is strongly connected with the periapse radius, increasing its value as the radius decreases. Nevertheless, the choice of this radius is limited, since it cannot be less than the planetary radius. Furthermore, there are other constrains depending on the flyby planet, which are tied to the characteristics of its atmosphere. The minimum values of the periapse altitude h_p ($r_p = R_P + h_p$) for the planets treated in this project, where R_P is the planet radius and r_p is the periapse radius, are shown on the

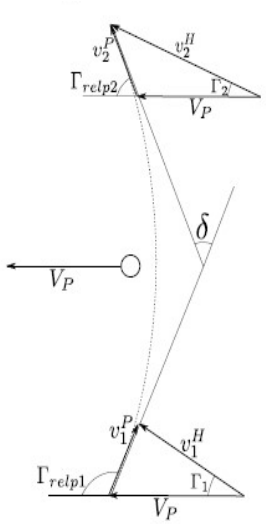

Figure 3.10 Velocity triangles for a Gravity Assist Maneuver, extracted from [3].

table 3.2.

Planet	**Minimum $\mathbf{h_p}$** (km)
Mars	250
Earth	1000
Jupiter	$9R_{\mathcal{Q}_+}$
Venus	300

Table 3.2 Minimum periapse altitude depending on the planet.

3.4.1 Powered maneuver: additional periapse impulse

Moreover, it is also possible to add an extra impulse to the Gravity Assist Maneuver. This additional impulse is conducted at periapsis. Assuming that the necessary extra-impulse at periapsis to reach a particular velocity is required, both velocities at periapsis (the arrival and the departure velocities) must be obtained:

$$\mathrm{V}_{1_{\mathrm{periapsis}}} = \mathrm{V}_{1\mathrm{ps}} = \sqrt{\frac{2\mu_{\mathrm{P}}}{\mathrm{r_p}} + (\mathrm{V}_1^{\mathrm{P}})^2}. \tag{3.12}$$

$$\mathrm{V}_{2_{\mathrm{periapsis}}} = \mathrm{V}_{2\mathrm{ps}} = \sqrt{\frac{2\mu_{\mathrm{P}}}{\mathrm{r_p}} + (\mathrm{V}_2^{\mathrm{P}})^2}. \tag{3.13}$$

The additional impulse needed at periapsis is given by:

$$\Delta\mathrm{V_p} = \mathrm{V}_{2\mathrm{ps}} - \mathrm{V}_{1\mathrm{ps}}. \tag{3.14}$$

The turn angle δ is calculated as the average of both turn angles, before and after the maneuver, due to the fact that the excess velocities are, in this case, different one from each other.

$$\delta_1 = 2\arcsin\left(1/\left(1+\frac{\mathrm{r_p}(\mathrm{V}_1^{\mathrm{P}})^2}{\mu_{\mathrm{P}}}\right)\right). \tag{3.15}$$

$$\delta_2 = 2\arcsin\left(1/\left(1+\frac{\mathrm{r_p}(\mathrm{V}_2^{\mathrm{P}})^2}{\mu_{\mathrm{P}}}\right)\right). \tag{3.16}$$

$$\delta = \frac{\delta_1+\delta_2}{2}. \tag{3.17}$$

3.5 Deep Space Maneuver (DSM)

Deep Space Maneuvers are additional impulses conducted by the spacecraft while travelling on the transfer orbit which modify the current trajectory. In this project, a n-DSM problem between two planets is analysed, thus $n+1$ intermediate orbits are considered. At the points where the DSMs are performed, there must be continuity in position, and it is considered that the velocity changes instantly. As mentioned above, the 2-D problem is studied, hence the third component of the impulse vector $\Delta\mathbf{V}$ is 0.

The procedure carried out consists on using the position of the impulses as a variable and propagating forward and backwards in time using Lambert's Problem. Therefore, at a point P_i, the velocity vectors $\mathbf{V}_i^-$ and $\mathbf{V}_i^+$ are, respectively, the velocity before and after the impulse. The impulse required for the Deep Space Maneuver, $\Delta\mathbf{V}$, is obtained as:

$$\Delta\mathbf{V} = \mathbf{V}_i^+ - \mathbf{V}_i^-. \tag{3.18}$$

4 Optimization Procedure

The aim of this chapter is to explain the optimization process developed in this project. Firstly, the general process, used for every different function considered, is introduced. Secondly, the optimisation tool is succinctly presented. Then, each specific problem studied is separately explained.

4.1 General procedure

In order to find an optimal solution for a specific mission, such as a direct mission from the Earth to another planet, the next procedure is followed.

1. **Data input**: Selection of the involved planets, the initial parking orbit around the departure planet, the final parking orbit around the arrival planet, the number of Deep Space Maneuvers or the number of Gravity Assist Maneuvers depending on the function evaluated.
2. **Initial guess**: It can be considered as the most important step. Since the optimization tool finds only a local optimal solution, its result is hugely dependent on the selected initial guess. The process to choose an initial guess is developed in chapter 5.
3. **Optimization parameters**: Definition of the linear and nonlinear constraints, as well as the lower and the upper bounds for every variable of the problem. The selection of the lower and the upper bounds for the departure and the arrival dates is paramount, due to the fact that they limit the space of possible solutions. The range of departure times is chosen consulting reference [3], where the most auspicious dates for each scenario have been analysed for the 3-D problem. Concerning the arrival dates, it must be borne in mind that the valid time interval of the Ephemeris ends by the year 2050.
4. **Optimization process of the function considered**: Using the Matlab tool *fmincon*, a local optimum for the desired *Objective Function* is searched. Generally, the *Objective Function* is the total mission impulse ΔV_{tot}.
5. **Results extraction**: It consists on obtaining all the variables of the problem for the optimal solution, in order to include them in an *excel* table, hence it is easier to compare different options.
6. **Representation of the optimal solution**: A Matlab plot is created to illustrate the orbits which have been chosen as well as to check their feasibility.

4.2 Optimization tool: *fmincon*

The optimization of interplanetary missions involves a huge number of variables and very complex functions. Moreover, these functions may have many different local optimal solutions, which are not the global optimum. Therefore, global optimization strategies such as Metaheuristic-based routines are commonly used due to the fact that they do not need a user-defined initial guess to start the optimization process. Nevertheless, their computation time is substantial.

In this project, the tool that has been chosen to search an optimal solution is *fmincon*, which is a Matlab function provided from the Optimization Toolbox, described in reference [15]. The main objective of this

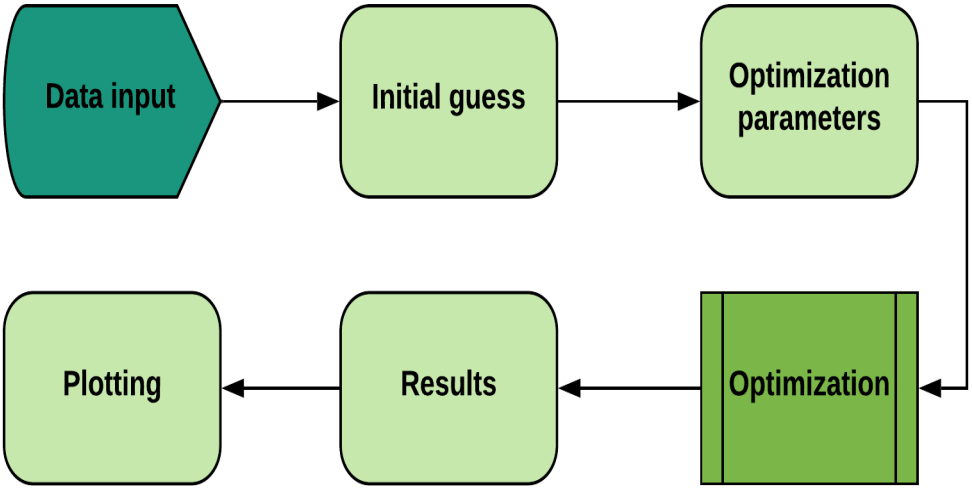

Figure 4.1 Diagram of the general procedure.

tool is to find the minimum value of a function, called *Objective Function*, given the constraints that must be satisfied. Tu put it another way, it is defined as a non-linear programming solver. The motive of using a local optimal searcher is to try to obtain equally feasible and optimal solutions as using a Metaheuristic-based algorithm such as a genetic algorithm but reducing the associated computation time.

The syntax of the function *fmincon* is simple, using $\mathbf{x}$ as the vector of the variables of the problem:

$$\min_{\mathbf{x}} \mathrm{f}(\mathbf{x}) \text{ such that } \begin{cases} \mathbf{c}(\mathbf{x}) \leq 0, \\ \mathbf{c}_{\mathrm{eq}}(\mathbf{x}) = 0, \\ \mathbf{A} \cdot \mathbf{x} \leq \mathbf{b}, \\ \mathbf{A}_{\mathrm{eq}} \cdot \mathbf{x} = \mathbf{b}_{\mathrm{eq}}, \\ \mathbf{lb} \leq \mathbf{x} \leq \mathbf{ub}, \end{cases} \tag{4.1}$$

where $\mathbf{c}(\mathbf{x})$ is the non-linear constraints vector, $\mathbf{c}_{\mathrm{eq}}(\mathbf{x})$ is the equality constraints vector, $\mathbf{A}$ and $\mathbf{b}$ are a real matrix and a real vector, respectively, which refer to the linear inequality constraints, $\mathbf{A}_{\mathrm{eq}}$ and $\mathbf{b}_{\mathrm{eq}}$ are a real matrix and a real vector, respectively, which refer to the linear equality constraints, **lb** and **ub** are vectors where the lower and upper bounds, respectively, are established for each variable. The required inputs include the *Objective Function* f, and an initial guess x_0 indeed. Furthermore, it is possible to set a wide range of optimization options like:

- Algorithm: The optimization algorithm used. The *Interior Point* Algorithm is selected since it can handle large and sparse problems, as well as small dense ones. It is a large-scale algorithm.
- Tolerances: For instance, it may be defined a tolerance for a constraint violation or for the step.
- Maximum number of evaluations or iterations: It is determined for the function evaluation and to define the maximum number of iterations that are allowed.
- Display: It provides an output in each iteration which shows information about the optimization procedure.

Eventually, the solution found by *fmincon* might not be a local optimal point, since the solver could have stopped searching due to a maximum number of iterations achieved. Therefore, the quality of the "optimal" solution obtained must be checked. The output *exitflag* makes it possible, returning a different number depending on the nature of the solution. For instance, if it has finished by achieving the maximum number of iterations it returns a 0 and if it has not found a feasible point it returns a -2.

4.3 Direct Mission

First, the direct mission involving only two impulses is analysed for each scenario. The mission begins from a parking orbit at an altitude h_{park1} over the departure planet's surface. The second impulse is performed at an altitude $h_{park2} = 9R_P$ over the destiny planet, where R_P is the radius of that planet. This decision was made in order to define the objective of each mission, which is setting the spacecraft at a circular orbit around the arrival planet. Even though That orbit's radius would be large enough to have a minor influence, it entails an arrival velocity limitation.

The algorithm implemented to solve the direct mission uses only the departure date (T_{ini}) and the arrival date (T_{end}) as variables:

1. Using the Ephemeris of both planets, their position and velocity at the desired time (T_{ini} for the departure planet and T_{end} for the destiny planet) are obtained.
2. Using Lambert tool, given the position of each planet, the departure and the arrival heliocentric velocities are calculated.
3. Using the heliocentric velocities and the planet's ones, the excess velocities and the parameter C_3 related to the departure and the arrival impulses are determined. Then, both impulses are calculated:

$$\Delta V_{ini} = \left| \sqrt{\frac{2\mu_{P_{ini}}}{r_{park1}} + C_{3_{ini}}} - \sqrt{\frac{\mu_{P_{ini}}}{r_{park1}}} \right|, \tag{4.2}$$

$$\Delta V_{end} = \left| \sqrt{\frac{2\mu_{P_{end}}}{r_{park2}} + C_{3_{end}}} - \sqrt{\frac{\mu_{P_{end}}}{r_{park2}}} \right|, \tag{4.3}$$

   where $\mu_{P_{ini}}$ and $\mu_{P_{end}}$ are the gravitational parameters of the departure planet and the arrival planet, respectively. $C_{3_{ini}} = V^2_{\infty_{ini}}$ refers to the departure planet and $C_{3_{end}} = V^2_{\infty_{end}}$, refers to the arrival planet.
4. Determination of the total impulse:

$$\Delta V_{tot} = \Delta V_{ini} + \Delta V_{end}. \tag{4.4}$$

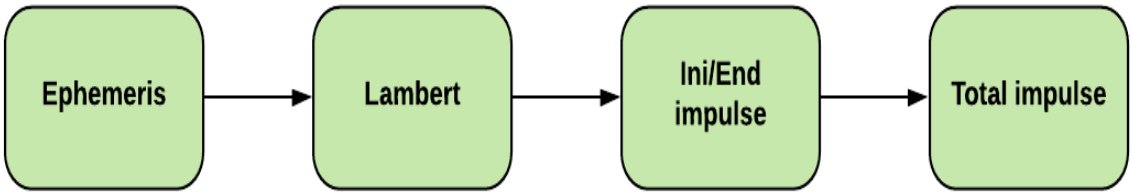

Figure 4.2 Schematic of the method developed for the direct mission.

4.3.1 Synodic Period

The synodic period (T_{syn}) is the amount of time necessary for the repetition of a relative configuration between two planets. In other words, how much time it takes to wait until a relative position of two planets recurs. It is calculated using the heliocentric period of the involved planets (T_{P_1} and T_{P_2}), which is defined as the time required to complete a revolution around the Sun.

$$T_{syn} = \frac{1}{|1/T_{P_1} - 1/T_{P_2}|}. \tag{4.5}$$

The synodic period becomes useful in order to consider the departure time window, hence the range of the departure time is set including a range of time similar to this period.

4.3.2 Pork Chop Plots

A common strategy used for the design of interplanetary missions is to represent the pork-chop plot. It consists on a diagram where contour lines of the mission parameter C_3 are represented as a function of the departure and arrival dates. The parameter C_3 is a measure of the specific energy, calculated from the excess velocity, V_∞, as $C_3 = V_\infty^2$. Thus, it is an appropriate way of comparing trajectories through the required energy as well as through its feasibility.

Generally, for the 3-D problem, if the orbits are elliptical and quasi-planar, there are two optimal trajectories, as it is shown in figure 4.3. The first one is known as a trajectory of Type I, faster than Hohmann trajectory ($\Delta\theta < 180$), and the other is known as a trajectory of Type II, slower than Hohmann's ($\Delta\theta > 180$). Even so, it should be noted that Hohmann trajectories only exist in the simplified case of circular planetary orbits. Nevertheless, since the 2-D problem involving elliptical and co-planar orbits is considered, both optimal possibilities may not exist. There might be more than one local minimum solutions, yet it is possible to find the best one, as it is explained in references [9] and [10]. In fact, the results calculated, which are shown in chapter 6, prove that one clear optimal solution could be identified.

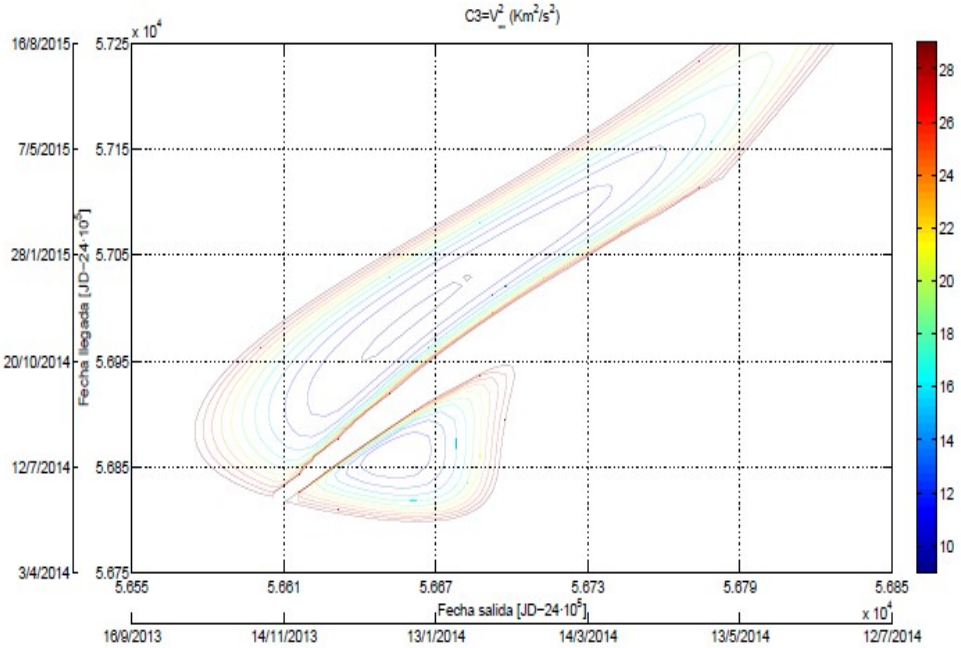

Figure 4.3 Pork chop plot for an Earth to Mars mission (3D problem), pulled from [3].

4.4 Multiple DSMs

Secondly, the mission with multiple Deep Space Maneuvers in addition to the departure and the arrival impulses is analysed. The required variables for this function are the departure date (T_{ini}) and the arrival date (T_{end}), as well as the date (T_{DSM_i}) and the position (r_{x_i} and r_{y_i}) where each Deep Space Maneuver is performed. As a result, the number of variables is increased as the number of DSM is extended.

The algorithm used to tackle this problem is simple:

1. Using the dates, the time of flight between each impulse is obtained. In addition, using the Ephemeris tool, the position of the planets is calculated.
2. Lambert tool is employed to obtain the heliocentric velocities at the beginning and the end of each segment, as it is shown in figure 4.4.
3. Using the heliocentric departure and arrival velocities, the excess speeds and the parameters C_3 are evaluated. Subsequently, the initial and final impulses are calculated using equations 4.2 and 4.3.
4. Calculation of the DSM impulses: looking at figure 4.4, having the heliocentric velocity before and after the i-DSM, it is possible to obtain the impulse required in that DSM (ΔV_{DSMi}) using equation 3.18.
5. Determination of the total impulse:

$$\Delta V_{tot} = \Delta V_{ini} + \Delta V_{end} + \sum_{i=1}^{i=n} \Delta V_{DSMi}. \tag{4.6}$$

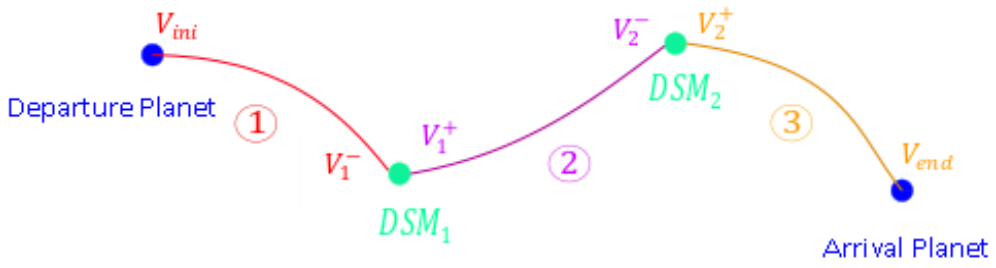

Figure 4.4 Schematic of the multiple DSM problem for the case of n=2.

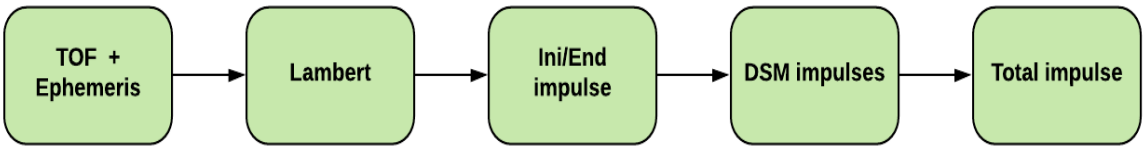

Figure 4.5 Diagram of the multiple DSM mission method.

As regards the optimisation parameters, the only special constraint, in addition to the fact that the DSM's dates must take place between the departure and the arrival dates, is associated with their position. The DSM position range is considered between the inferior planet orbit and 1.2 times the superior planet orbit radius.

4.5 GAM

The next stage analysed is a mission which involves a Gravity Assist Maneuver (GAM). A powered maneuver is considered by adding an extra impulse at the periapsis as seen by the flyby planet. The method requires as variables the departure date (T_{ini}), the arrival date (T_{end}), the flyby date (T_{GAM}) and the periapse radius of the hyperbola (r_p) where the extra-impulse is performed.

The algorithm developed is based on the ideas that are shown on reference [6]. The match point is the Gravity Assist Maneuver, thus the trajectory is propagated forward in time from the start point, which is the departure planet, until that match point. Then, the trajectory is propagated backwards in time from the end point, which is the destiny planet, to the match point. There, at the match point, the constraints are applied. The procedure consists on:

1. Forward propagation: using planetary Ephemeris and Lambert tool, the departure heliocentric velocity V_{ini} and the heliocentric velocity when the spacecraft arrives at the flyby planet V^{+}_{GAM} are determined.
2. Backwards propagation: using planetary Ephemeris and Lambert tool, the arrival heliocentric velocity at the destiny planet V_{end} and the heliocentric velocity when the spacecraft leaves the flyby planet V^{-}_{GAM} are calculated.
3. Using V_{ini}, the excess velocity and its associated parameter C_3 are obtained. Accordingly, the initial impulse is calculated using equation 4.2.
4. Using V_{end}, the excess velocity and its associated parameter C_3 are determined. Then, the final impulse is calculated using equation 4.3.
5. Applying the method described in Section 3.4, the required extra-impulse at periapsis ΔV_p is obtained via V^{+}_{GAM} and V^{-}_{GAM}.
6. Determination of the total impulse:

$$\Delta V_{tot} = \Delta V_{ini} + \Delta V_{end} + \Delta V_p. \tag{4.7}$$

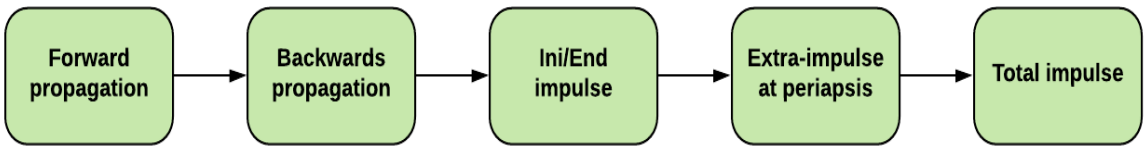

Figure 4.6 Diagram of the method implemented to solve the Gravity Assist Maneuver problem.

With respect to the optimization parameters, there is a minimum periapse radius which depends on the planet under consideration.

4.5.1 Multiple GAMs (MGA)

Including more than one Gravity Assist Maneuver is also studied. Consequently, the number of variables depends on the number of maneuvers, adding the periapse radius and the date of each additional GAM to the variables already defined. In that case, all the propagation is done forward in time. Hence using Lambert tool and the Ephemeris of each pair of planets, the velocities are calculated, and then, the extra-impulses at periapsis are determined for each flyby.

4.6 DSMs in the GAM mission

Finally, a problem which involves both DSMs and GAM is considered is solved. It consists on a mission to a destiny planet via another planet flyby, including the possibility of adding n DSMs in the first transfer and m DSMs in the second one. The variables of this problem are the departure date (T_{ini}), the arrival date (T_{end}), the flyby date (T_{GAM}), the periapse radius of the hyperbola associated with the flyby, as well as the date and the position of every DSM included.

1. Multiple DSM from the departure planet to the flyby planet. The following parameters are obtained:
   - Initial impulse : ΔV_{ini}.
   - Every DSM impulse of the first phase: $\Delta V_{DSM1_1}, \ldots, \Delta V_{DSM1_n}$.
   - Heliocentric velocity when the spacecraft arrives at the flyby planet V_{GAM}^{-}.
2. Multiple DSM from the flyby planet to the destiny planet. The following parameters are determined:
   - Heliocentric velocity when the spacecraft leaves the flyby planet V_{GAM}^{+}.
   - Every DSM impulse of the second phase: $\Delta V_{DSM2_1}, \ldots, \Delta V_{DSM2_m}$.
   - Final impulse : ΔV_{end}.
3. Gravity Assist Maneuver: calculation of the required extra-impulse at periapsis ΔV_p, via V_{GAM}^{+} and V_{GAM}^{-}.
4. Determination of the total impulse:

$$\Delta V_{tot} = \Delta V_{ini} + \Delta V_{end} + \sum_{i=1}^{i=n} \Delta V_{DSM1_i} + \sum_{j=1}^{j=m} \Delta V_{DSM2_j} + \Delta V_p. \tag{4.8}$$

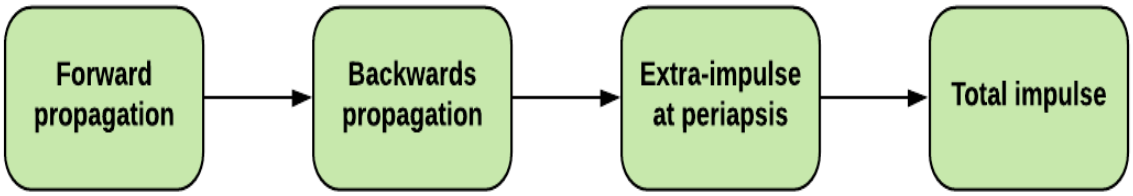

Figure 4.7 Diagram of the procedure developed to solve a problem which involves a Gravity Assist Maneuver including multiple Deep Space Maneuvers in both phases.

5 Initial Guess Algorithms

Assuming that the optimization process is carried out using the Matlab tool *fmincon*, the selection of the initial guess is paramount. As a matter of fact, *fmincon* tries to find a local minimum of the considered function, that search extremely depends on the initial point used. In this project, different ways of selecting the initial guess are evaluated in order to compare them. Algorithm 1 requires a previous investigation to find a suitable initial guess, yet it is associated with a short computation time. Algorithm 2 does not need a preceding analysis but explores a wide range of possible initial guesses, hence its computational burden increases. Lastly, Algorithm 3 includes an Ant Colony Optimization procedure allowing to focus on the best solutions previously found. Therefore, the last algorithm can improve the solutions obtained by Algorithm 2.

5.1 Algorithm 1

This first method consists on inferring the initial solution of the considered problem using a previous simpler problem. Hence the optimal solution of a basic problem associated with the one under analysis must be evaluated. Then, that solution is used as the initial guess to solve the complex problem. To illustrate this method, the procedure followed to determine the initial guess for each variable of every problem is explained.

5.1.1 DSMs

In this case, the variables under consideration are T_{ini}, T_{end}, as well as T_{DSM_i}, r_{x_i} and r_{y_i}, with $i = 1,\ldots,n$, for each DSM. The initial guess for each variable is determined using this method:

- $T_{ini}^{[0]}$ and $T_{end}^{[0]}$: Using the obtained optimal departure and arrival dates for the direct problem.
- $T_{DSM_i}^{[0]}$: The time interval $T_{ini}^{[0]} - T_{end}^{[0]}$ is divided into $n+1$ sections of the same length, t_{gap}. The initial guess for the date of the i-DSM is $T_{DSM_i}^{[0]} = T_{ini}^{[0]} + i \cdot t_{gap}$.
- $r_{x_i}^{[0]}$ and $r_{y_i}^{[0]}$: Dividing the distance between both planets involved equidistantly, into $n+1$ sections of length d_{gap}. The initial guess for the components of the position r_x and r_y of the i-DSM is $r_{x|y_i}^{[0]} = L_{Pini} \pm i \cdot d_{gap}$, where L_{Pini} is the mean distance from the Sun to the initial planet. The sign depends on whether the vehicle is going to a superior or to an inferior planet. After having set these initial values, if the *Objective Function* is not defined at the evaluated point or its value is very high, the sign and the value of each variable $r_{x_i}^{[0]}$ and $r_{y_i}^{[0]}$ are randomly modified.

5.1.2 GAM

- $T_{ini}^{[0]}$: Using the optimal departure date for the direct transfer to the first flyby planet.
- $T_{GAM}^{[0]}$: If it is the first GAM, it is calculated as the optimal arrival date for the direct transfer from the departure planet. In case of multiple flybys, the date of each GAM is obtained dividing the problem into only 1 GAM problems and using their optimal dates, employing previously solved problems or adding the respective Hohmann's times of flight between each planet. For instance, when studying a problem Earth-Mars-Earth-Saturn, the simpler problems Earth-Mars-Earth and Mars-Earth-Saturn are analysed

to obtain an initial guess for the complete problem. However, if the mission Earth-Mars-Saturn have already been studied, it might be used the optimal solution for that problem to set the initial guess for the date of the Mars flyby and then the time of flight of the Hohmann transfer Mars-Earth may be added to calculate an initial guess for the Earth flyby.

- $T_{end}^{[0]}$: Adding the time of flight of the Hohmann transfer between the last flyby planet and the destiny planet to the date of the last flyby.
- $r_p^{[0]}$: It is set as 1.1 times the minimum periapse radius of each flyby planet.

5.1.3 DSMs + GAM

The method consists on a combination of both the DSMs problem and the GAM problem. Each initial guess is obtained using the previously mentioned procedure.

5.2 Algorithm 2

The foremost problem of Algorithm 1 is the fact that the optimization solver finds only the closest local minimum to the initial guess. Nevertheless, as the problem is highly non-linear, thus probably non-convex, there may be many different local minimum points in the search space. Accordingly, if a local minimum is found, other possible solutions are left out of consideration. One way to overcome this problem in order to augment the chances of computing the global minimum, is to calculate multiple local minimums providing the local optimization solver with different initial guesses. After this, they are compared, and the best solution is selected. Moreover, the initial guesses must be diverse so that the found local optimums would be different. The main strength of this algorithm is the fact that it explores myriad varied possibilities.

Once having determined the search space for these two variables, T_{ini} and T_{end}, each range is divided equidistantly (a same number of divisions for each variable is not compulsory). The next stage is combining them in order to create different pairs of initial guesses, and then obtaining the rest of the variables which are required for each pair. Obviously, this method's computation time is higher than the one associated with Algorithm 1, yet it makes possible to discover new optimal points.

Figure 5.1 illustrates an example of the procedure implemented by Algorithm 2, taking only two initial dates and three final dates (denoted as 2-3). As it is shown, if any of the obtained initial guesses are not feasible, in this case due to the fact that the arrival date is previous to the departure one, they are eliminated.

The method is focused on calculating several initial guesses via the departure time T_{ini} and the arrival time T_{end}. The rest of the variables of the problem for the case of multiple DSM are obtained following the same procedure developed in Algorithm 1.

Exceptionally, when using this algorithm on Gravity Assist Maneuvers problems, the initial guesses for the dates of the flybys are calculated following this method, depending on the number of GAMs:

- 1 GAM case: for each pair of T_{ini} and T_{end}, the time interval between them is divided into 4 gaps, obtaining 3 intermediate dates. These 3 dates are used as initial guesses for the date of the GAM, hence each pair of T_{ini} and T_{end} produces 3 different initial guesses.
- 2 GAM case: for each pair of T_{ini} and T_{end}, the time interval between them is divided into 5 gaps, obtaining 4 intermediate dates. The first 2 dates are employed as initial guesses for the date of the first GAM. Then, the last 2 dates are used as initial guesses for the date of the second GAM. After this, they are combined, thus for each pair of T_{ini} and T_{end}, 4 different initial guesses are determined.
- 3 GAM case: for each pair of T_{ini} and T_{end}, the time interval between them is divided into 7 gaps obtaining 6 intermediate dates. The first 2 dates are used as initial guesses for the date of the first GAM, dates 3 and 4 are employed as initial guesses for the date of the second GAM, and the last 2 dates are used as initial guesses for the date of the third GAM. A couple of different initial guesses is obtained for each pair of T_{ini} and T_{end}. The first involves the intermediate dates 1, 3 and 5 and the second, dates 2, 4 and 6.

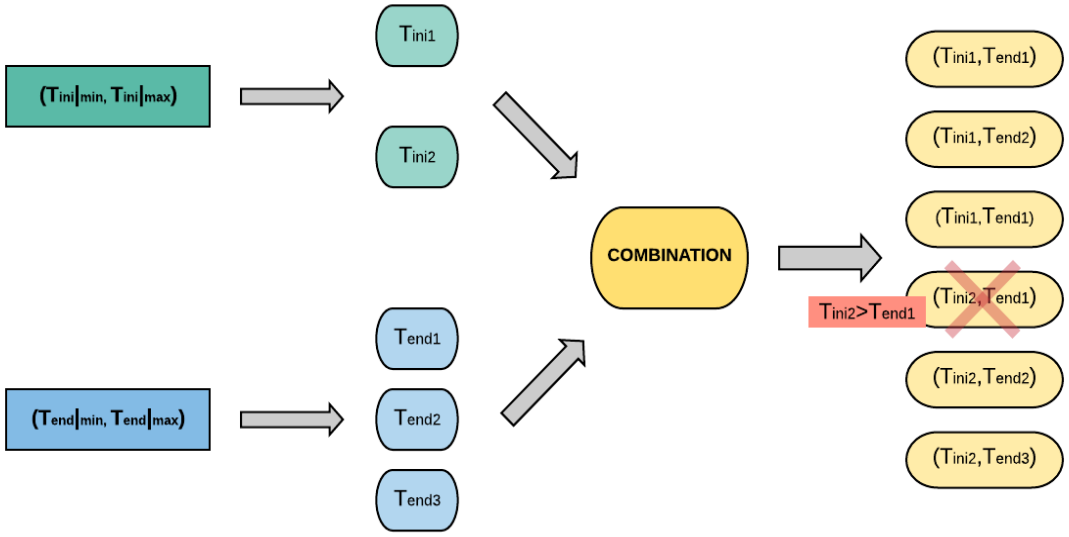

Figure 5.1 Illustration of the method developed in Algorithm 2.

5.3 Algorithm 3

A challenging problem which arises when applying Algorithm 2 is the lack of feedback. To put it another way, after having obtained the results via Algorithm 2, it would make sense to exploit the best solutions in order to enhance them. An alternative approach to tackle this problem is to include a Metaheuristic-based Ant Colony Optimization (ACO) adapted to continuous optimization.

As explained in reference [4], the inspiring source of ACO is the foraging behaviour of real ants when searching for food. First, they aimlessly explore the area surrounding their nest. When an ant discovers a food source, it evaluates it and carries a portion back to the nest leaving a pheromone trail on the ground during the return trip. The amount of pheromone deposited depends on the quality and the quantity of the food source, which has been previously evaluated. Therefore, it has been proved that indirect communication among ants enables them to discover the shortest path between their nest and food sources. This capability of real ant colonies has inspired the definition of algorithms that can solve hard combinatorial optimization problems, such as the one presented in this project.

The overall idea of ACO is the incremental construction of solutions based on the biased probabilistic choice of solutions components, determined by the pheromone. At each construction step, ants make a probabilistic choice of the solution component. Moreover, ants sample a probability density function (PDF) for each variable which defines the probability of selecting a certain value for that variable. A Gaussian kernel PDF is used due to the fact that it can identify more than one promising solution, as it may have more than one maximum. A Gaussian kernel function is represented in figure 5.2. For the variable i of the problem whose vector of variables is $\mathbf{x}$, it is denoted as $G^i(\mathbf{x})$.

$$G^i(\mathbf{x}) = \sum_{l=1}^{k} \omega_l g_l^i(\mathbf{x}) = \sum_{l=1}^{k} \omega_l \frac{1}{\sigma_l^i \sqrt{2\pi}} e^{-\frac{(\mathbf{x}-\mu_l^i)^2}{2(\sigma_l^i)^2}}. \tag{5.1}$$

There must be defined as many Gaussian kernel PDFs as the number of problem variables (i = 1,...,n) and the design parameter k represents the number of ants used to look for a solution (necessarily $k > n$). Each of the PDF is parameterized with three vectors:

- μ^i is the vector of means. In each iteration, the solutions which were obtained and ordered in the previous iteration become the means of the actual one.

$$\mu^i = \{\mu_1^i, \ldots, \mu_l^i\} = \{s_1^{i-1}, \ldots, s_l^{i-1}\}. \tag{5.2}$$

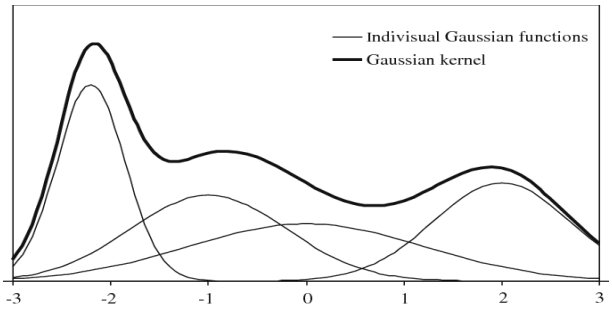

Figure 5.2 Example of three Gaussian functions and their superposition: the resulting Gaussian kernel. Extracted from [5].

- ω is the vector of weights associated with the individual Gaussian functions. Each solution is evaluated and ranked, thus they are sorted according to their rank. For instance, solution s_l has rank l. The weight ω_l of the solution s_l is calculated as:

$$\omega_l = \frac{1}{qk\sqrt{2\pi}} e^{-\frac{(l-1)^2}{2q^2k^2}}, \tag{5.3}$$

  which substantially defines the weight to be a value of the Gaussian function with argument l, mean 1.0 and standard deviation qk. The design parameter q is associated with the diversity of the solutions built. When its value is small, the best-ranked solutions are strongly preferred.

- σ^i is the vector of standard deviations. The value of the standard deviation σ_l^i at construction step i is determined using the average distance from the chosen solution s_l to other stored solutions and the parameter ξ.

$$\sigma_l^i = \xi \sum_{e=1}^{k} \frac{\left|s_e^i - s_l^i\right|}{k-1}. \tag{5.4}$$

  The parameter $\xi > 0$ is associated with the pheromone evaporation, in other words, it determines how fast solutions are forgotten. The higher the value of ξ, the lower the convergence speed of the algorithm.

Summarising, the procedure followed by the Ant Colony Algorithm consists on constructing solutions using a PDF and keeping the best results throughout all the iterations. Consequently, it returns k ordered solutions, with the optimal solution in the first place.

With this in mind, it is easy to explain the procedure developed in Algorithm 3. First, the initial guesses are obtained as it is done in Algorithm 2 and *fmincon* is employed to calculate the associated local optimal solutions. Then, they are rated and ordered depending on their rank. Subsequently, the Ant Colony Algorithm begins, having a number of loops established. Eventually, the best 20 solutions that are found are used as initial guesses for *fmincon* and, after that, the optimal one is selected.

Finally, the effect of the design parameters, q and ξ, has been evaluated. The selection process for the initial guesses (Algorithm 2) involves the exploration of the search space. Provided that, the real need when applying ACO is to focus on the discovered solutions and to try to improve them. Therefore, the parameter q, related to the diversity of the solutions constructed, may have a small value, such as 0.1 or 0.3. Notwithstanding, different values of q have been tested for each problem under analysis, and the one that provides the best solution has been selected for each problem. After having tested the effect of the parameter $\xi > 0$, it has been concluded that it must be close to 0. Due to the fact that the unit of time employed is Julian Days, which entails the use of huge numbers, as well as the need to obtain solutions which meet the requirements set as constraints, the created solutions may not excessively vary. Thus, the value of ξ is set as 0.01. The selection of the value of k depends on the problem under consideration and it is given by the process of creating initial guesses via Algorithm 2.

6 Simulation Scenarios and Optimization Results

In order to test Section 5 algorithms, an objective mission must be defined. Three different scenarios are analysed: firstly, a mission to Mars (where no GAM is explored due to its Earth proximity); secondly, a mission to Saturn; and finally, a mission to Jupiter. The processor Intel Core i7-5500U/2.40GHz is employed. It must be clarified that the main objective in every mission is to reduce the total cost (fuel consumption), hence it would be mentioned that an algorithm performs better than another if it achieves a solution with a lower total impulse.

6.1 Mission to Mars

The first considered scenario is a mission to Mars. A huge number of space missions to Mars have been developed in the last 50 years, but there are also many projects about future missions to the "Red Planet" as well, like ExoMars in 2022. One of the most important reasons of such considerable activity is the fact that Mars is the closest planet to the Earth, thus the energy needed to complete this type of mission is easily available using the current technology. What's more, Mars is a frequent target not only for its proximity to the Earth, but also for the possibility of having extraterrestrial life. The mission analysed in this section departs from a parking orbit at an altitude $h_{park1} = 200$ km over the Earth and arrives at another parking orbit at an altitude $h_{park2} = 9R_{♂}$ over Mars. It is important to remind that the symbol ♂ refers to Mars.

6.1.1 Direct Mission

First, the direct transfer is studied. Using reference [3], it is decided to search an optimal solution where the departure date might take place in the second half of 2016. Then, the Earth-to-Mars synodic period is obtained, $T_{syn} = 780$ days. It is slightly longer than two years, hence the range of departure time T_{ini} lies between 01/11/2025 and 01/02/2028.

To obtain an initial flight time estimation for the Earth-Mars transfer, in order to set the range of the arrival date, the time of the Hohmann transfer using their mean distances to the Sun is calculated (about 260 days). Since different types of trajectories must be compared, faster and slower than Hohmann's, the range of the final date is established between 01/06/2026 and 01/01/2030.

After that, the pork-chop plot associated with this mission is built. Although there seems to be a couple of local minimum points, the optimal one can be clearly identified. The optimal departure date is around November 2026 and the optimal arrival date takes place in September 2027.

To test the solver *fmincon*, the following initial guess is employed: $T_{ini} = 01/11/2026$, $T_{end} = 01/09/2027$. The solver correctly finds the optimal solution shown on figure 6.1, which is detailed on table 6.1 and figure 6.2.

The solution obtained is similar to a Hohmann transfer, having a transfer angle close to 180° and a transfer orbit which looks like half of an ellipse. In a nutshell, when searching for an optimal solution using a two-impulse maneuver, it seems that the best solution is always a kind of Hohmann transfer orbit.

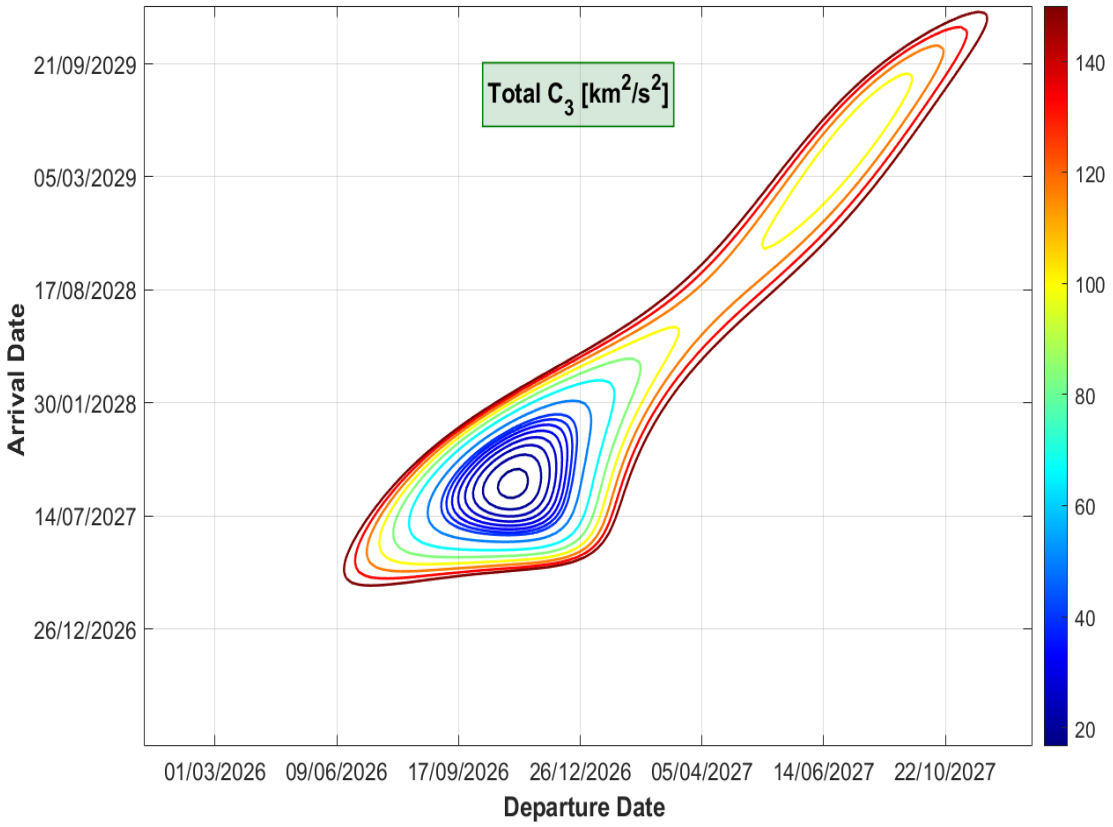

Figure 6.1 Pork chop plot: Direct mission to Mars.

	Departure Date	Arrival Date	ΔV_{total} (km/s)	ΔV_{ini} (km/s)	ΔV_{end} (km/s)
Direct Mission	31/10/2026	06/09/2027	5.500449	3.622519	1.877930

TOF (days)	Computation time (s)
311	7.551940

Table 6.1 Optimal solution for the direct mission to Mars.

6.1.2 Deep Space Maneuvers

Even though nowadays the total impulse required for the direct mission seems feasible, every available option to reduce the cost of a space project must be studied. For that reason, the effect of including multiple Deep Space Maneuvers, in addition to the departure and arrival impulses, is evaluated.

The case of multiple Deep Space Maneuvers is analysed without a time of flight limitation. In order to apply Algorithm 1, it is necessary to set an initial guess for the multiple DSM problem. The chosen initial guess is the optimal one calculated for the direct transfer. The range of the initial time is established as (01/01/2026,01/01/2028) and the range of the arrival date as (01/01/2027,01/01/2035). Then the results using every Algorithm are represented on tables 6.2 and 6.3. For Algorithm 2, (5-5) represents the number of initial and final dates employed, respectively.

In light of the obtained results, Algorithm 2 definitely outperforms Algorithm 1 although its computation time is remarkably higher though not prohibitive. Moreover, the use of Algorithm 3 allows to slightly enhance the solution calculated by Algorithm 2, significantly increasing the computation time. Nevertheless, Algorithm 2 and Algorithm 3 do not require an initial procedure in order to find a feasible initial guess. The solutions found applying Algorithm 3 are presented on table 6.4 and in figure 6.3.

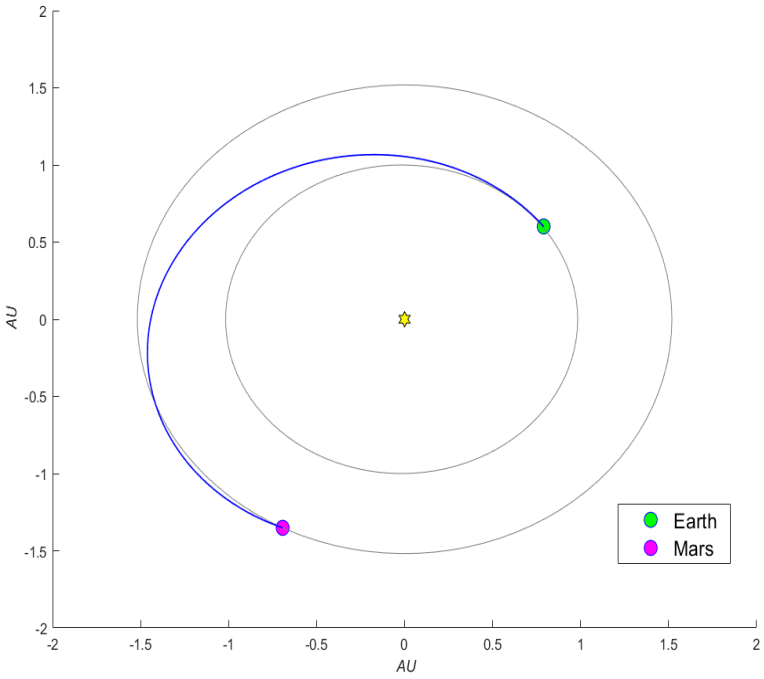

Figure 6.2 Optimal solution for the direct mission to Mars.

	ΔV_{total} (km/s)	TOF (days)	Computation time (s)
Algorithm 1	5.502706	309	4.648377
Algorithm 2 (5-5)	5.226575	833	43.004496
Algorithm 3 (k=46, q=0.1)	5.222407	828	148.646799

Table 6.2 Optimal solution for the mission to Mars via 1 DSM.

	ΔV_{total} (km/s)	TOF (days)	Computation time (s)
Algorithm 1	5.870755	338	12.994341
Algorithm 2 (5-5)	5.246494	857	112.563479
Algorithm 3 (k=46, q=0.3)	5.239120	807	200.397695

Table 6.3 Optimal solution for the mission to Mars via 2 DSMs.

	Departure Date	Arrival Date	ΔV_{ini} (km/s)	ΔV_{end} (km/s)	ΔV_{DSM1} (km/s)
n=1	06/08/2026	11/11/2028	3.783566	1.278002	0.160839
n=2	30/07/2026	14/10/2028	3.772061	1.340180	0.126279

DSM_1 Date	ΔV_{DSM2} (km/s)	DSM_2 Date
06/05/2027	/	/
13/05/2027	$2.672 \cdot 10^{-7}$	01/01/2028

Table 6.4 Optimal solution for the mission to Mars via 1 and 2 DSMs via Algorithm 3.

The optimal procedure for a multiple DSM problem involves more than one revolution around the Sun in order to reduce the final braking impulse. Therefore, we conclude that including multiple Deep Space Maneuvers decreases the cost of a mission and it has been shown that adding more than two intermediate impulses may not be worthy. For a mission to Mars, the optimal fuel consumption solution includes one DSM, reducing the total cost of the mission by 5% and increasing the time of flight by 500 days. As it is shown on table 6.4, when searching an optimal solution including 2 DSMs, the solver concluded that the best option is not to add a second DSM.

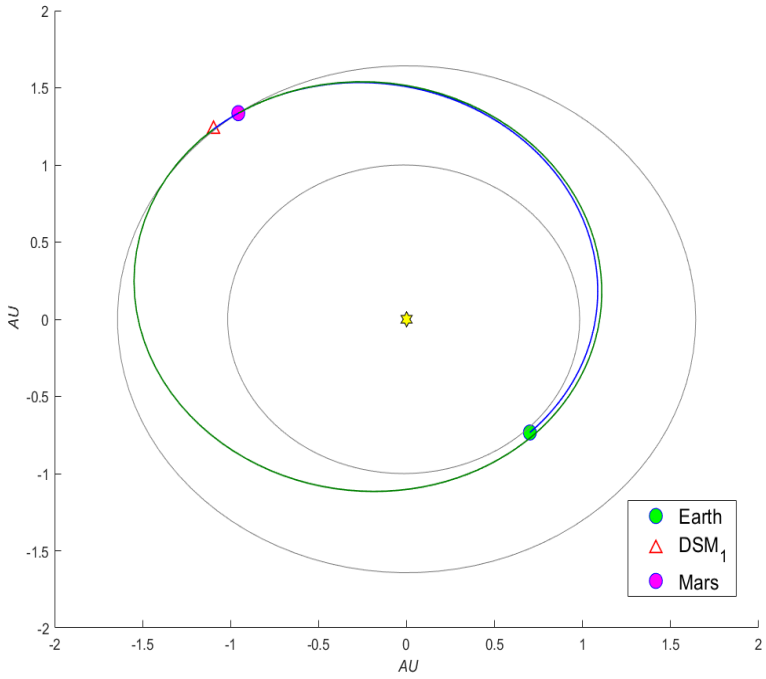

Figure 6.3 Optimal solution for the mission to Mars via 1 DSM using Algorithm 3.

6.2 Mission to Saturn

The next scenario consists on an Earth-Saturn mission, which is a considerably further objective than Mars, thus GAMs are considered. Saturn was initially visited with the aim of performing flyby maneuvers, in missions like Pioneer 11 and Voyager. Then, NASA's Cassini spacecraft orbited the Ringed Planet for the first time in 2004. Some of the reasons that make Saturn an attractive exploration objective is the fact that it has more than 50 known natural satellites as well as the interest of studying its complex ring system. The mission investigated in this section starts from a parking orbit at an altitude $h_{park1} = 200$ km over the Earth and reaches another parking orbit at an altitude $h_{park2} = 9R_{\text{♄}}$ around Saturn. It is important to remind that the symbol ♄ refers to the planet Saturn.

6.2.1 Direct Mission

First, the direct transfer is analysed. Using reference [3], it is decided to search an optimal solution where the departure date takes place in 2025. Then, the Earth-Saturn synodic period is calculated, $T_{syn} = 378$ days, which is approximately a year. Thus, the range of the departure time T_{ini} is set between 01/01/2025 and 01/01/2026.

The time of flight necessary to go from the Earth to Saturn is estimated using the time of the Hohmann transfer. It is approximately 6 years. Since different types of trajectories must be contrasted, faster and slower than Hohmann's, the range of the final time is established between 01/06/2026 and 01/01/2040.

The next step is building the pork-chop plot associated with this mission. The optimal solution can be clearly identified. It involves a departure date around the month of May of 2025 and an arrival date around May 2031.

After that, the solver *fmincon* is used to search the optimal point shown in the pork-chop plot, operating with the initial guess ($T_{ini} = 01/05/2025$, $T_{end} = 01/05/2031$). The solver successfully finds the optimal point shown in figure 6.4, which is detailed on table 6.5 and in figure 6.5:

	Departure Date	Arrival Date	ΔV_{total} (km/s)	ΔV_{ini} (km/s)	ΔV_{end}(km/s)
Direct Mission	30/05/2025	15/02/2031	11.939133	7.289887	4.649246

TOF (days)	Computation time (s)
2087	3.171922

Table 6.5 Optimal solution for the direct mission to Saturn.

In this case, likewise it happened in the mission to Mars, the optimal solution found is a kind of Hohmann transfer. However, the cost of the mission has been extremely increased, especially due to the huge initial impulse required. The reason of such a large initial impulse is the immense distance between the Earth and Saturn. Consequently, it would be beneficial to find an alternative way to reach Saturn.

6.2.2 Deep Space Maneuvers

After that, the case of multiple Deep Space Maneuvers is studied without a time of flight limitation. In order to apply Algorithm 1, it is necessary to set an initial guess which is the optimal one calculated for the direct transfer. The range of the initial time is established between 01/01/2025 and 01/01/2026 and the range of the arrival date lies between 01/06/2026 and 01/01/2050. Then the results using all the Algorithms are represented on tables 6.6 and 6.7.

	ΔV_{total} (km/s)	TOF (days)	Computation time (s)
Algorithm 1	11.948046	2172	8.760772
Algorithm 2 (3-10)	11.708306	6963	75.985829
Algorithm 3 (k=50, q=0.3)	11.707129	6986	190.234830

Table 6.6 Optimal solution for the mission to Saturn via 1 DSM.

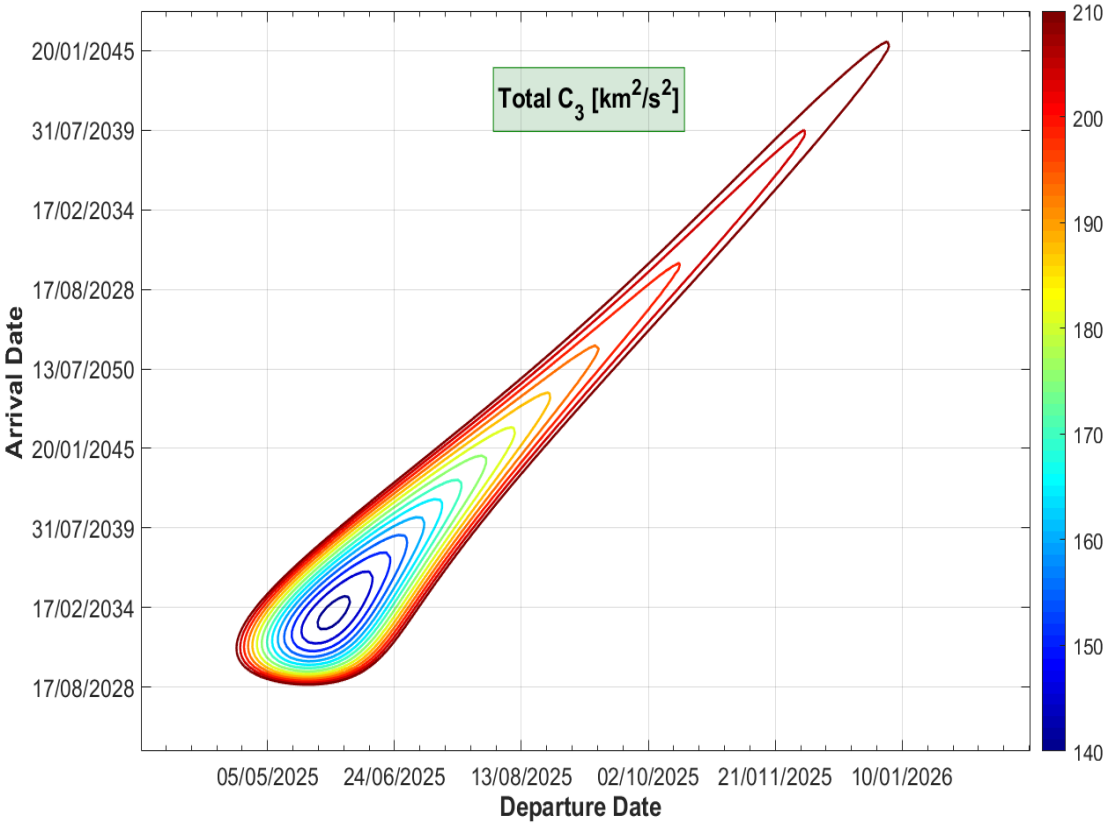

Figure 6.4 Pork chop plot: Direct mission to Saturn.

	ΔV_{total} (km/s)	TOF (days)	Computation time (s)
Algorithm 1	11.948413	2172	33.886305
Algorithm 2 (3-10)	11.965802	6105	261.231823
Algorithm 3 (k=50, q=0.1)	11.948408	2172	638.306790

Table 6.7 Optimal solution for the mission to Saturn via 2 DSMs.

In view of the results, it appears that Algorithms 2 and 3 can discover a better solution in terms of the cost than Algorithm 1. Particularly, the optimal solutions using Algorithm 3 are presented on table 6.8. It has been found that the optimal solution is not to include DSMs, since the impulses are below tolerance (reaching mm/s accuracy would be unfeasible for a DSM thus these impulses are considered as 0), and it involves a complete revolution around the Sun (figure 6.6). That solution could not be obtained in Section 6.2.1 since the Lambert's Problem tool does not have the possibility of calculating multiple revolution orbits.

	Departure Date	Arrival Date	ΔV_{ini} (km/s)	ΔV_{end} (km/s)	ΔV_{DSM1} (km/s)	DSM_1 Date	ΔV_{DSM2} (km/s)	DSM_2 Date
n=1	25/11/2025	10/01/2045	7.272395	4.434734	$4.520 \cdot 10^{-8}$	22/01/2035	/	/
n=2	01/06/2025	13/05/2031	7.292580	4.655827	$8.193 \cdot 10^{-7}$	07/05/2027	$3.702 \cdot 10^{-7}$	10/05/2029

Table 6.8 Optimal solution for the mission to Saturn via 1 and 2 DSMs via Algorithm 3.

In brief, the cost of the direct transfer has not been reduced enough. Particularly, the initial impulse is actually large. In conclusion, these impulses are expensive to be performed, hence another alternative method

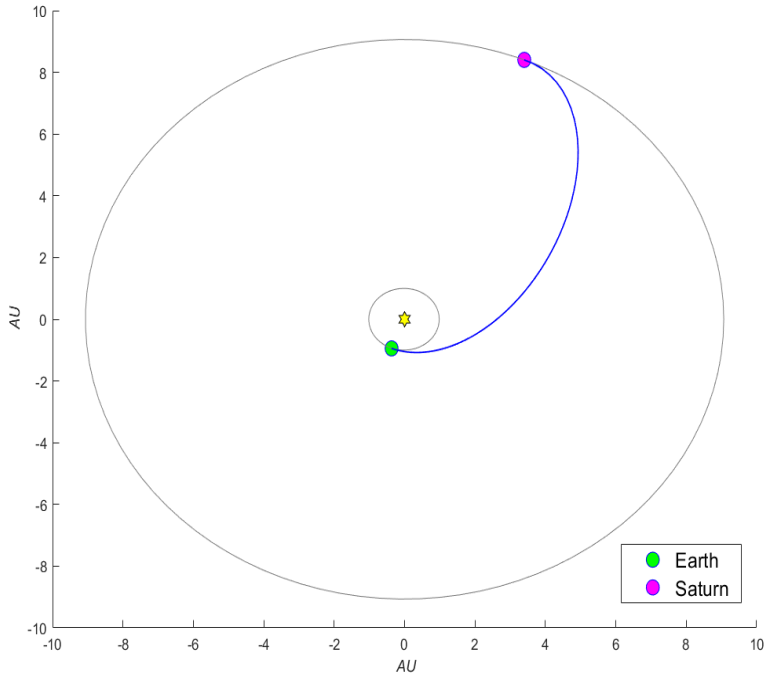

Figure 6.5 Optimal solution for the direct mission to Saturn.

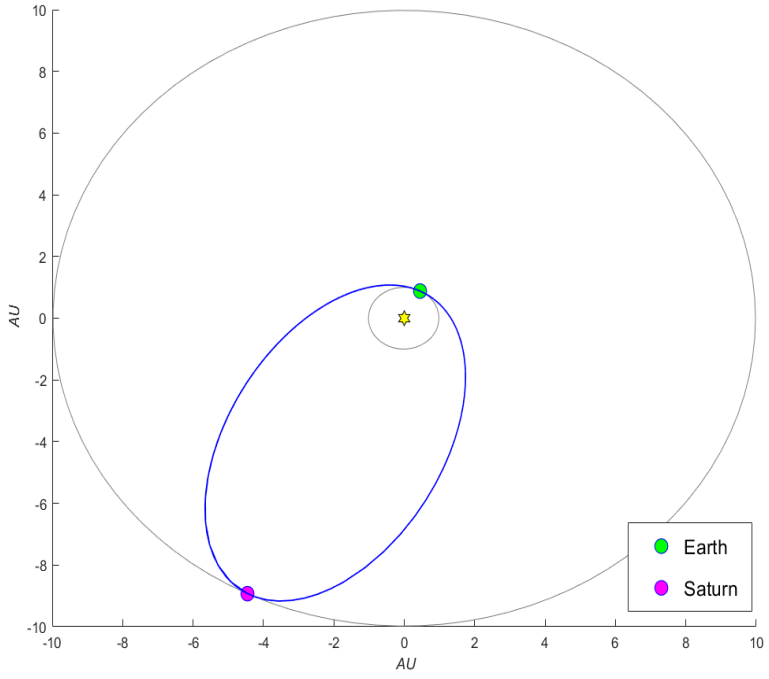

Figure 6.6 Optimal solution for the mission to Saturn using Algorithm 3.

must be found to make this mission possible.

6.2.3 Gravity Assist Maneuvers

The next stage, looking forward to reducing the mission cost, is using a Gravity Assist Maneuver. This may increase the time of flight needed, but hopefully it could diminish the total impulse required. First, diverse cases are separately analysed, and then they are all compared. Generally, the range for the initial time lies between 01/01/2024 and 01/01/2026, since it is the required wider range comparing Jupiter's and Mars' flybys.

Jupiter flyby

The first case under analysis is a mission to Saturn via swing by Jupiter. To apply Algorithm 1, the direct transfer Earth-Jupiter must be evaluated, to obtain an initial flight time estimation. The Earth-Jupiter synodic period is $T_{syn} = 399$ days. Consulting reference [3], it is decided to search an optimal departure date around September 2025. Using *fmincon*, an optimal solution for the direct mission to Jupiter is detected for the C_3 parameter (figure 6.18) as the interest lies in the flyby over Jupiter. The optimal point is almost a Hohmann transfer with a flight time of 1056 days, leaving the Earth on 25 September 2025 and arriving at Jupiter on 16 august 2028. Using these dates, as well as adding 10 years for the trajectory between Jupiter and Saturn, which is the time of flight of a Hohmann transfer between them, the initial guess is determined: $T_{ini} = 01/09/2025$, $T_{GAM} = 01/08/2028$, $T_{end} = 01/08/2038$.

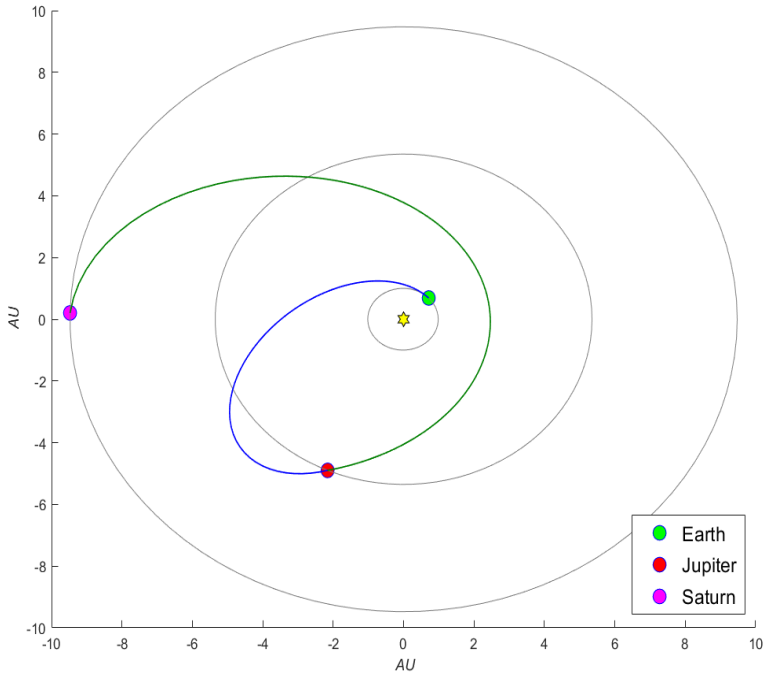

Figure 6.7 Optimal solution for the mission to Saturn via GAM around Jupiter using algorithm 3.

To apply Algorithms 2 and 3, the range of possible arrival dates must be set. Having calculated both times of flight associated with the Hohmann transfer for Earth-Jupiter (6 years) and for Jupiter-Saturn (10 years), the range of the arrival date is set as (01/01/2035, 01/01/2045). Then, the optimal solution using 3 divisions for T_{ini} and 5 for T_{end} is calculated using Algorithm 2.

The result is almost the same using each algorithm as it is represented on table 6.9. Although the computation time is extremely larger using Algorithm 3, a similar solution is obtained with every tool. However, Algorithm 1 requires a previous analysis of the direct mission individually, which is not necessary to apply Algorithm 2 and Algorithm 3. In brief, including a Gravity Assist Maneuver reduces the total impulse required although it increases the time of flight of the mission. The optimal fuel consumption solution found using Algorithm 3, which reduces the cost of the direct mission by 13% and increased the time of flight by

	ΔV_{total} (km/s)	TOF (days)	Computation time (s)
Algorithm 1	10.418022	4956	2.554828
Algorithm 2 (3-5)	10.417692	4976	59.048049
Algorithm 3 (k=150, q=0.3)	10.417683	4980	404.416760

Table 6.9 Comparison of the optimal solution for the mission to Saturn via GAM around Jupiter obtained using each algorithm.

2900 days, is presented on table 6.10 and figure 6.7, where the symbol ♃ refers to the planet Jupiter.

Departure Date	Arrival Date	GAM Date	ΔV_{ini} (km/s)	ΔV_{end} (km/s)	ΔV_{GAM} (km/s)
06/11/2025	27/06/2039	07/11/2030	6.603505	3.814178	$3.730 \cdot 10^{-8}$

$r_p/R_♃$	δ (º)
10.5	97.717

Table 6.10 Optimal solution for the mission to Saturn via GAM around Jupiter using Algorithm 3.

Mars flyby

Secondly, a Gravity Assist Maneuver around Mars is considered. The optimal solution for the Earth-Mars direct transfer is already known but in a different time interval. Consequently, it is necessary to search an optimal point within the range of dates established. It involves a departure date around August of 2025, an arrival date at Mars around July of 2027 and a time of flight of about 700 days. Calculating the time of flight of the Hohmann transfer for Mars-Saturn, 2390 days (6.5 years) and adding it to the date of the Mars flyby, the initial guess for the arrival date at Saturn is obtained (January 2033).

Having calculated both times of flight associated with the Hohmann transfer for Earth-Mars (260 days) and for Mars-Saturn (6.5 years), the range of the arrival date is established as (01/01/2028, 01/01/2040).

	ΔV_{total} (km/s)	TOF (days)	Computation time (s)
Algorithm 1	10.059794	2703	1.878261
Algorithm 2 (3-5)	9.197015	2954	74.539424
Algorithm 3 (k=150, q=0.3)	9.080567	2846	426.528380

Table 6.11 Comparison of the optimal solution for the mission to Saturn via GAM around Mars obtained using each algorithm.

In this case, each algorithm improves the result obtained by the previous one. Algorithm 2 clearly outperforms Algorithm 1, allowing to achieve a total impulse 9% lower. Algorithm 3 performs better than Algorithm 2, slightly reducing the total impulse on an extra 1,2% although the computation time is 6 times longer. The optimal solution found using Algorithm 3 is presented on table 6.12 and figure 6.8

Departure Date	Arrival Date	GAM Date	ΔV_{ini} (km/s)	ΔV_{end} (km/s)	ΔV_{GAM} (km/s)
07/09/2025	26/03/2033	20/09/2027	4.708733	4.371782	$5.2330 \cdot 10^{-5}$

h_{mars} (km)	δ (º)
290.1054	16.477

Table 6.12 Optimal solution for the mission to Saturn via GAM around Mars obtained using Algorithm 3.

The results obtained show better results with a Mars flyby than with a Jupiter flyby, since not only the mission's cost has been reduced but also the time of flight required. The initial impulse has been diminished by 35% due to the fact that Mars is closer to the Earth than Jupiter.

Mars flyby + Jupiter flyby

At this point, it is time to include multiple Gravity Assist Maneuvers in the mission. Having studied both the individual fly by Jupiter and by Mars, the first idea which comes to our mind is a mission involving a couple of

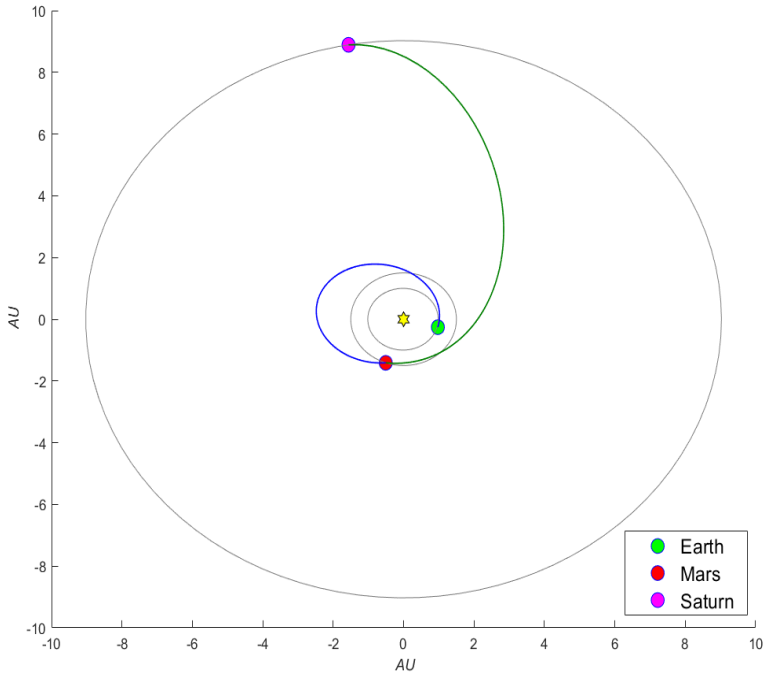

Figure 6.8 Optimal solution for the mission to Saturn via GAM around Mars using Algorithm 3.

flybys, the first one around Mars and the second one around Jupiter. The initial guesses for the departure date and the date of the Mars flyby are taken from the optimal solution of the previous problem, 02/11/2025 and 12/01/2028 respectively. Having the required time of flight for a Hohmann transfer between Mars and Jupiter, approximately 3 years, the initial guess for the date of the fly by Jupiter is set, 01/01/2031. Finally, as the time of the Hohmann transfer from Jupiter to Saturn is 10 years, the initial guess for the arrival date is 01/01/41.

Then, it must be established the range of possible arrival dates to apply Algorithms 2 and 3. Since the approximate total flight time adding all the Hohmann's times of flight is 14 years, the range set is (01/01/2035, 01/01/2050).

	ΔV_{total} (km/s)	TOF (days)	Computation time (s)
Algorithm 1	14.550293	5538	1.488648
Algorithm 2 (3-8)	7.74480682	5699	270.368512
Algorithm 3 (k=96, q=0.3)	7.744707	5700	423.423840

Table 6.13 Comparison of the optimal solution for the mission to Saturn via 2 GAMs (around Mars and Jupiter) obtained using each algorithm.

Algorithms 2 and 3 definitely perform better than Algorithm 1 in this case, achieving a solution with half of the cost associated with the solution found by Algorithm 1, which does not reduce the cost of the direct mission.

Departure Date	Arrival Date	$GAM_{♂}$ Date	$GAM_{♄}$ Date	ΔV_{ini} (km/s)	ΔV_{end} (km/s)
16/02/2025	25/09/2040	11/07/2026	12/12/2031	4.048778	3.686066

$\Delta V_{GAM_{♂}}$ (km/s)	$\Delta V_{GAM_{♄}}$ (km/s)	$h_{p♂}$ (km)	$r_{p♄}/R_{♄}$	$\delta_{♂}$ (º)	$\delta_{♄}$ (º)
$3.605 \cdot 10^{-4}$	$9.5016 \cdot 10^{-3}$	301.7373	10.1	11.806	105.917

Table 6.14 Optimal solution for the mission to Saturn via 2 GAMs (around Mars and Jupiter) obtained using Algorithm 3.

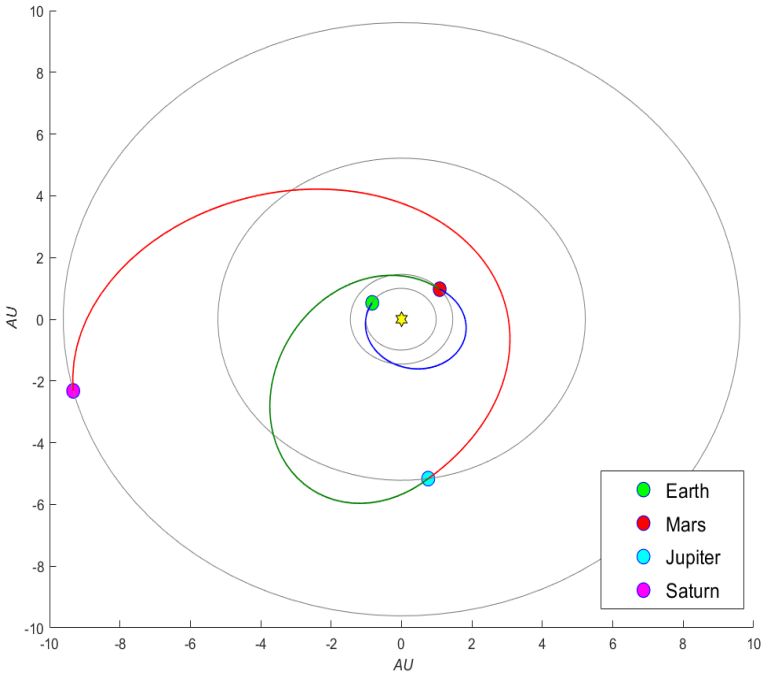

Figure 6.9 Optimal solution for the mission to Saturn via 2 GAMs (around Mars and Jupiter) using Algorithm 3.

The optimal solution found by Algorithm 3 improves the ones which were calculated using just one GAM in terms of the cost, even though it requires a higher time of flight. Moreover, the total impulse entails a 35% reduction of the cost of the direct mission. It seems that a combination of two flybys, one around a planet close to the departure one and another one around a planet "close" to the destiny involves a reduction of both the initial and the final impulse, respectively.

Mars flyby + Earth flyby

This time, a mission to Saturn which involves a Gravity Assist Maneuver around Mars and then another one around the Earth is analysed. The problem is divided to obtain an initial guess for Algorithm 1. First, a mission with a fly by Mars which then comes back to the Earth is analysed. Then, a mission from Mars to Saturn via an Earth flyby is studied, using as initial guess the optimal arrival date at Mars of the previous mission, then adding the time of flight of the Hohmann trajectories between Mars and the Earth, and the one associated with an Earth-Saturn trajectory. The results obtained for both missions are shown on table 6.15.

	Departure Date	Arrival Date	GAM Date	TOF (days)	Computation time (s)
Earth-Mars-Earth	15/03/2025	29/08/2029	22/08/2026	1627	4.135338
Mars-Earth-Saturn	10/08/2026	16/06/2033	02/07/2027	2502	1.502232

Table 6.15 Optimal solution for previous missions required.

After having examined these missions, it is possible to set an approximate initial guess for the complete problem using Algorithm 1. The initial date for the initial guess is 15/03/2025, the date of the Mars flyby, 15/08/2026, the date of the second flyby, 01/01/2028, and the arrival date at Saturn, 01/01/2034. For the final date, as the approximate total time of flight adding the time of every Hohmann transfer is 7.5 years, the range set is (01/01/2030, 01/01/2040).

As it is shown on table 6.16, once again Algorithm 2 outperforms Algorithm 1 and it is surpassed by Algorithm 3. The optimal solution achieved by Algorithm 3 reduces the cost of the direct mission by 21%, yet the time of flight is increased by only 810 days. All the details of that optimal solution are shown on table 6.17 and figure 6.12.

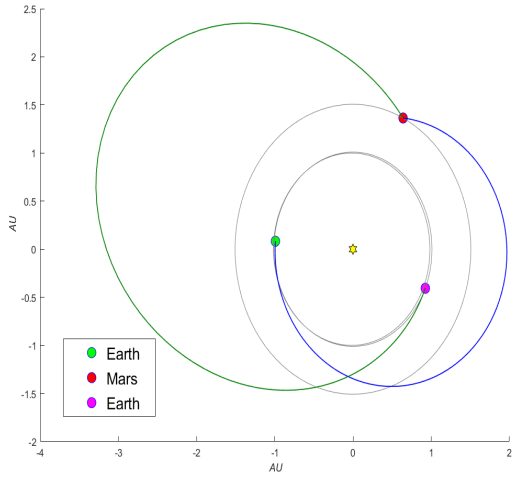

Figure 6.10 Optimal solution for the mission Earth-Mars-Earth using algorithm 1.

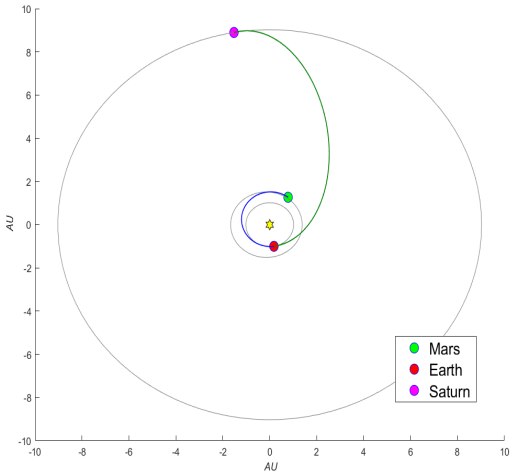

Figure 6.11 Optimal solution for the mission Mars-Earth-Saturn using algorithm 1.

	ΔV_{total} (km/s)	TOF (days)	Computation time (s)
Algorithm 1	11.221134	3139	3.433734
Algorithm 2 (3-8)	9.700413	4735	305.911223
Algorithm 3 (k= 96, q=0.6)	9.383823	2897	426.168110

Table 6.16 Comparison of the optimal solution for the mission to Saturn via 2 GAMs (around Mars and around the Earth) obtained using each algorithm.

Departure Date	Arrival Date	$GAM_{♂}$ Date	$GAM_{⊕}$ Date	ΔV_{ini} (km/s)	ΔV_{end} (km/s)
07/03/2025	10/02/2033	15/08/2026	09/06/2028	4.162791	4.863154

$\Delta V_{GAM_{♂}}$ (km/s)	$\Delta V_{GAM_{⊕}}$ (km/s)	$h_{p♂}$ (km)	$h_{p⊕}$ (km)	$\delta_{♂}$ (º)	$\delta_{⊕}$ (º)
$3.103 \cdot 10^{-5}$	0.357846	267.0842	1074.5597	11.358	105.917

Table 6.17 Optimal solution for the mission to Saturn via 2 GAMs (around Mars and around the Earth) calculated using Algorithm 3.

Mars flyby + Earth flyby + Jupiter flyby

Following, a mission which involves 3 Gravity Assist Maneuvers (Mars, the Earth and Jupiter) is studied. In order to apply Algorithm 1, it is possible to use the optimal solution of the problem with two flybys (Mars and Earth) as initial guess for the initial time (06/03/2024), the time of the Mars flyby (31/08/2026) and the date of the Earth flyby (23/07/2031). For the rest of the variables, a mission from the Earth to Saturn via a Jupiter flyby is analysed, using (23/07/2031) as the initial value for the departure time for Algorithm 1. As that mission have already been studied, even though it was analysed for different dates, the required time of flight for that evaluated mission is used to estimate the date of the fly by Jupiter (adding 1800 days to the date of the Earth flyby) and the arrival date (adding 3100 days to the date of the Jupiter flyby).

Then, with respect to Algorithms 2 and 3, the range of the possible arrival dates is set, knowing that the total time adding the Hohmann transfer's time of flight of every trajectory involved is approximately 14 years, as (01/01/2035, 01/01/2050).

The best solution in terms of the cost is reached by Algorithm 3, decreasing the total impulse of the direct mission by 22.5%. That solution is presented on table 6.19 and figure 6.13. The time of flight of this solution has been remarkably increased, as it is more than 18 years longer than the time of the direct transfer.

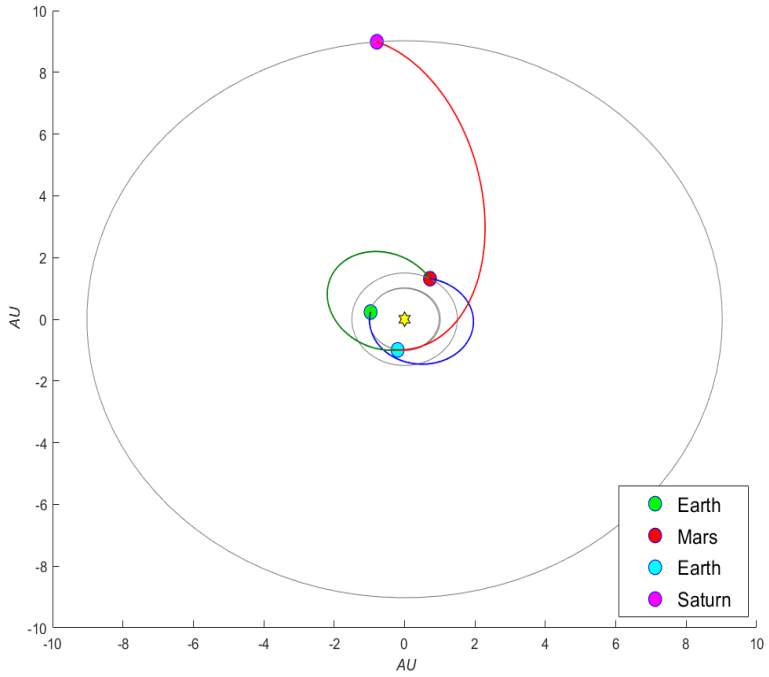

Figure 6.12 Optimal solution for the mission to Saturn via 2 GAMs (around Mars and around the Earth) calculated using Algorithm 3.

	ΔV_{total} (km/s)	TOF (days)	Computation time (s)
Algorithm 1	12.190059	7595	1.847411
Algorithm 2 (3-8)	9.471824	8267	213.900810
Algorithm 3 (k= 80, q=0.3)	9.242740	8920	443.750960

Table 6.18 Comparison of the optimal solution for the mission to Saturn via 3 GAMs (around Mars, the Earth and Jupiter) obtained using each algorithm.

Departure Date	Arrival Date	$GAM_{♂}$ Date	$GAM_{⊕}$ Date	$GAM_{♃}$ Date	ΔV_{ini} (km/s)
16/03/2024	17/08/2048	22/09/2026	03/10/2033	25/10/2041	5.077950

ΔV_{end} (km/s)	$\Delta V_{GAM_{♂}}$ (km/s)	$\Delta V_{GAM_{⊕}}$ (km/s)	$\Delta V_{GAM_{♃}}$ (km/s)	$h_{p♂}$ (km)	$h_{p⊕}$ (km)
3.942244	$8.403 \cdot 10^{-4}$	$1.675 \cdot 10^{-4}$	0.221538	288.4388	1074.6673

$r_{p♃}/R_{♃}$	$\delta_{♂}$ (º)	$\delta_{⊕}$ (º)	$\delta_{♃}$ (º)
10.10	6.105	40.914	80.704

Table 6.19 Optimal solution for the mission to Saturn via 3 GAMs (around Mars, the Earth and Jupiter) calculated using Algorithm 3.

Venus flyby + Mars flyby + Jupiter flyby

Subsequently, another mission which involves 3 Gravity Assist Maneuvers (Venus, Mars and Jupiter) is studied. To apply Algorithm 1, the time of flight of the Hohmann transfer from the Earth to Venus (146 days) and from Venus to Mars (217 days) are calculated. To determine an initial solution for the departure time, the direct transfer Earth-Venus is analysed. The optimal initial date found is 01/01/2025. Then, adding the respective times of flight, it is calculated the date of the Venus flyby (01/06/2025), the date of the Mars flyby (01/02/2026), the date of the Jupiter flyby (01/02/2029) and the arrival date (01/01/2040).

The range of the possible final dates is lying between 01/01/2035 and 01/01/2050, since the total time adding the Hohmann transfer's time of flight of every trajectory involved is approximately 14 years.

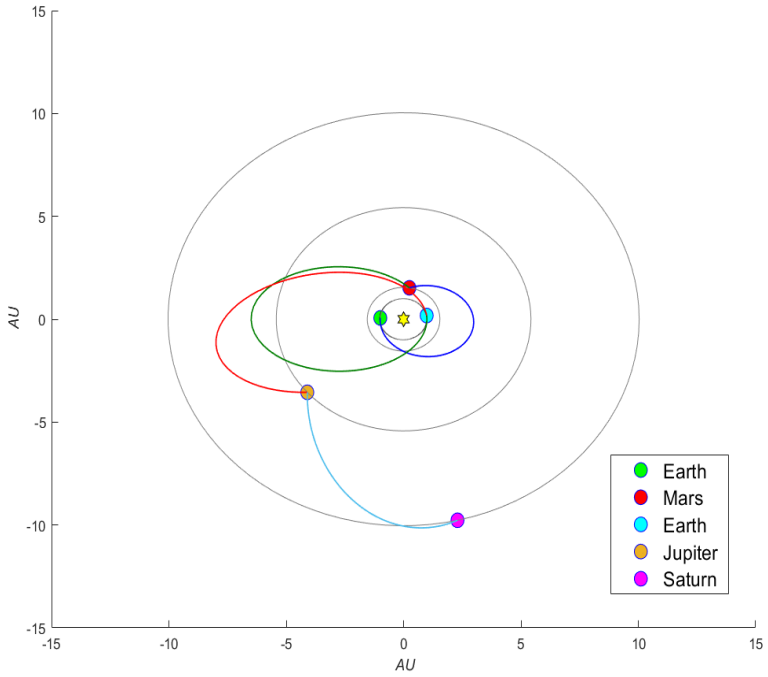

Figure 6.13 Optimal solution for the mission to Saturn via 3 GAMs (around Mars, the Earth and Jupiter) calculated using Algorithm 3.

	ΔV_{total} (km/s)	TOF (days)	Computation time (s)
Algorithm 1	17.429050	5477	2.334620
Algorithm 2 (5-10)	8.521050	5240	441.926640
Algorithm 3 (k= 100, q=0.6)	7.986568	4939	555.826420

Table 6.20 Comparison of the optimal solution for the mission to Saturn via 3 GAMs (around Venus, Mars and Jupiter) obtained using each algorithm.

In this case, Algorithm 1 does not perform well at all, because the initial guess may be inaccurate. However, using Algorithm 3 an exceptional solution has been found, reducing by 33% the cost of the direct mission. That solution is presented on table 6.21 and figure 6.14.

Departure Date	Arrival Date	$GAM_{♀}$ Date	$GAM_{♂}$ Date	$GAM_{♄}$ Date	ΔV_{ini} (km/s)
22/03/2025	30/09/2038	07/09/2025	05/03/2028	30/04/2030	4.081123

ΔV_{end} (km/s)	$\Delta V_{GAM_{♀}}$ (km/s)	$\Delta V_{GAM_{♂}}$ (km/s)	$\Delta V_{GAM_{♄}}$ (km/s)	$h_{p♀}$ (km)	$h_{p♂}$ (km)
3.893304	$3.490 \cdot 10^{-4}$	$1.707 \cdot 10^{-3}$	0.010086	536.9137	667.2086

$r_{p♄}/R_{♄}$	$\delta_{♀}$ (º)	$\delta_{♂}$ (º)	$\delta_{♄}$ (º)
10.0992	40.479	3.736	94.088

Table 6.21 Optimal solution for the mission to Saturn via 3 GAMs (around Venus, Mars and Jupiter) calculated using Algorithm 3.

An important factor which leads to the reduction of the total impulse is the decrease of the initial impulse, due to the proximity of Venus to the Earth. By the same token, the GAM around Jupiter allows to reduce the final impulse.

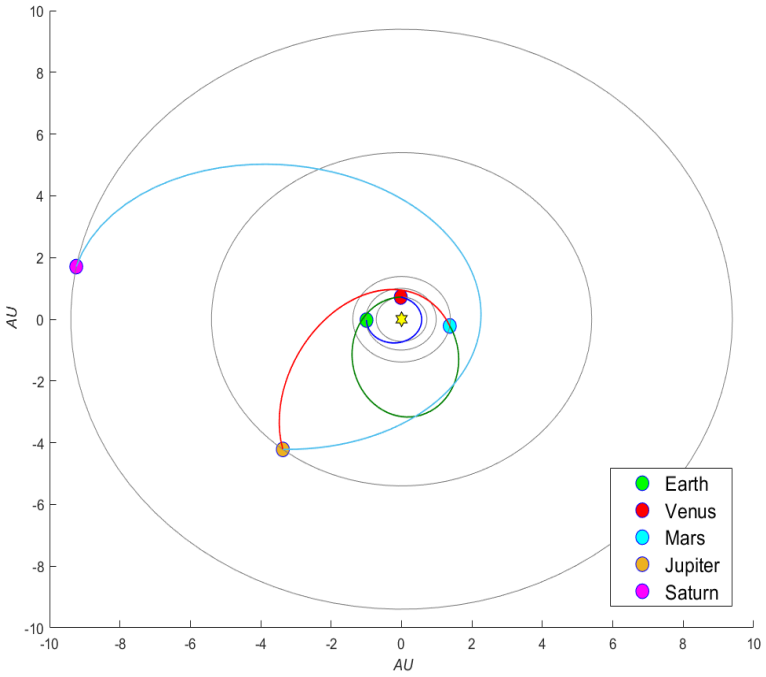

Figure 6.14 Optimal solution for the mission to Saturn via 3 GAMs (around Venus, Mars and Jupiter) calculated using Algorithm 3.

Venus flyby + Earth flyby + Jupiter flyby

Finally, a mission which involves 3 Gravity Assist Maneuvers (Venus, the Earth and Jupiter) is analysed. To apply Algorithm 1, the optimal solution from the Earth to Venus previously calculated is employed. The optimal initial date found is 01/01/2025. Then, adding the respective times of flight, the following dates are calculated: the date of the Venus flyby (1/06/2025), the date of the Earth flyby (01/01/2026), the date of the Jupiter flyby (01/01/2029) and the arrival date (01/01/2039).

After that, the range of the possible arrival dates is set as (01/01/2035, 01/01/2050).

	ΔV_{total} (km/s)	TOF (days)	Computation time (s)
Algorithm 1	13.322511	5108	3.760382
Algorithm 2 (5-10)	10.108138	6082	431.139110
Algorithm 3 (k= 100, q=0.6)	8.883265	8408	641.068890

Table 6.22 Comparison of the optimal solution for the mission to Saturn via 3 GAMs (around Venus, the Earth and Jupiter) obtained using each algorithm.

The results shown on table 6.22 perfectly define the utility of each algorithm. In order to achieve a better solution, it is necessary to spend a higher computation time. In this case, that time is not extremely longer for Algorithm 3 than for Algorithm 2. The solution calculated by Algorithm 3 is presented on table 6.23 and figure 6.15.

The maneuvers included in this mission allow to reduce the cost of the direct mission by 25% although the time of flight is remarkably increased. Again, the impulse at Jupiter's periapsis reduces the final impulse.

Departure Date	Arrival Date	GAM$_{\mathrm{♀}}$ Date	GAM$_{\oplus}$ Date	GAM$_{♄}$ Date	ΔV_{ini} (km/s)
25/04/2025	01/05/2048	11/10/2025	15/12/2034	30/10/2041	4.599033

ΔV_{end} (km/s)	$\Delta V_{GAM_{\mathrm{♀}}}$ (km/s)	$\Delta V_{GAM_{\oplus}}$ (km/s)	$\Delta V_{GAM_{♄}}$ (km/s)	$h_{p\mathrm{♀}}$ (km)	$h_{p\oplus}$ (km)
3.969878	$2.7212 \cdot 10^{-3}$	$3.3590 \cdot 10^{-3}$	0.308274	361.8965	1072.1820

$r_{p♄}/R_{♄}$	$\delta_{\mathrm{♀}}$ (º)	$\delta_{\oplus}$ (º)	$\delta_{♄}$ (º)
10.10	28.591	13.342	78.43374

Table 6.23 Optimal solution for the mission to Saturn via 3 GAMs (around Venus, the Earth and Jupiter) calculated using Algorithm 3.

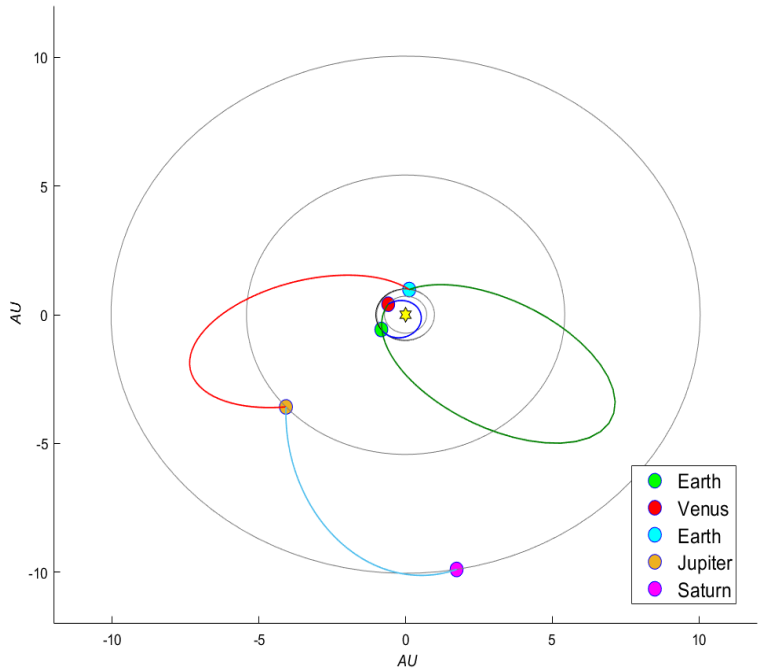

Figure 6.15 Optimal solution for the mission to Saturn via 3 GAMs (around Venus, the Earth and Jupiter) calculated using Algorithm 3.

Final comparison

Finally, to clarify the results of each algorithm in order to properly compare them, they are all included in a table.

	ΔV_{total} (km/s)	TOF (days)	ΔV_{ini} (km/s)	ΔV_{end} (km/s)
Direct	11.9391	2087	7.2899	4.6492
Jupiter flyby	10.4180	4956	6.6028	3.8152
Mars flyby	10.0598	2703	5.6054	4.4530
Mars + Jupiter	14.5503	5538	9.7417	3.9667
Mars + Earth	11.2211	3139	4.3984	4.7462
Mars + Earth + Jupiter	12.1901	7595	5.0511	3.4755
Venus + Mars + Jupiter	17.4291	5476	6.1980	6.4203
Venus + Earth + Jupiter	13.3225	5108	3.9141	5.2398

Table 6.24 Optimal solution for every GAM mission to Saturn using Algorithm 1.

	ΔV_{total} (km/s)	TOF (days)	ΔV_{ini} (km/s)	ΔV_{end} (km/s)
Direct	11.9391	2087	7.2899	4.6492
Jupiter GAM	10.4177	4976	6.6034	3.8143
Mars GAM	9.1970	2954	4.7594	4.4376
Mars + Jupiter	7.7448	5699	4.0488	3.6861
Mars + Earth	9.7004	4735	5.0447	4.6541
Mars + Earth + Jupiter	9.4718	8267	5.6517	3.7973
Venus + Mars + Jupiter	8.5210	5240	4.3563	4.1525
Venus + Earth + Jupiter	10.1081	6082	4.7151	3.8824

Table 6.25 Optimal solution for every GAM mission to Saturn using Algorithm 2.

	ΔV_{total} (km/s)	TOF (days)	ΔV_{ini} (km/s)	ΔV_{end} (km/s)
Direct	11.9391	2087	7.2899	4.6492
Jupiter GAM	10.4177	4980	6.6035	3.8142
Mars GAM	9.0806	2846	4.7087	4.3718
Mars + Jupiter	7.7447	5700	4.0488	3.6861
Mars + Earth	9.3838	2897	4.1628	4.8632
Mars + Earth + Jupiter	9.2427	8920	5.0780	3.9422
Venus + Mars + Jupiter	7.9866	4939	4.0811	3.8933
Venus + Earth + Jupiter	8.8833	8408	4.5990	3.9699

Table 6.26 Optimal solution for every GAM mission to Saturn using Algorithm 3.

Gravity Assists make possible to enormously reduce the cost of the mission even though they increase the necessary time of flight. Generally, Algorithm 2 provides better solutions than Algorithm 1 as it can explore more than one local minimum. Then, Algorithm 3 enhances the solutions found by Algorithm 2, obtaining better results. Looking at table 6.26, there seems to be an optimal configuration for the mission to Saturn, which is a mission that involves two Gravity Assist Maneuvers, the first around Mars and the second around Jupiter. The required total impulse has been reduced by 35% from the direct mission and the time of flight is about 10 years longer. Moreover, the total cost of the mission would be just 32,5% more expensive than the cost associated with the mission to Mars. If we are interested on reducing the time of flight, a mission which involves three GAMs (Venus, the Earth and Jupiter) is 760 days shorter, yet it only reduces the cost of the direct mission by 33%. Another interesting option, which does not excessively increase the time of flight of the direct mission, is a mission with a swing by Mars which reduces by 24% the total cost of the direct mission and takes "just" about 750 days more than the direct one. For every gravity assist, not only the total cost of the mission is reduced, but also the initial impulse required at the Earth's parking orbit.

6.2.4 DSM + GAM

Finally, a mission which involves a Gravity Assist Maneuver and Deep Space Maneuvers in each of their phases is analysed. Both cases, a fly by Jupiter and a fly by Mars, are studied with n DSMs in the first phase (Earth to flyby planet) and m DSMs on the second one (flyby planet to Saturn).

This time, the method followed to solve the problem with each Algorithm has been modified due to the complexity of the problem. Once *fmincon* has found a local optimal solution, it is used as an initial guess to run *fmincon* again, until a feasible solution is obtained. Talking about Algorithm 2, the process to obtain an initial guess using the range of the times defined is a combination between the method used to solve the GAM problem and the one applied for the DSM problem. Furthermore, in view of the previous results, the initial guess for the position of the intermediate impulses is closer to the destiny planet.

Jupiter flyby

The initial guess for the dates required by Algorithm 1, is the optimal solution without Deep Space Maneuvers which was obtained using Algorithm 3. Then, the range of the arrival time is extended until 01/01/2050 to apply Algorithms 2 and 3.

	ΔV_{total} (km/s)	TOF (days)	Computation time (s)
Algorithm 1	10.418505	4976	2.554828
Algorithm 2 (3-10)	10.174892	5831	211.577279
Algorithm 3 (k=90, q=0.3)	9.924585	6437	464.935560

Table 6.27 Comparison of the optimal solution for the mission to Saturn via flyby around Jupiter including 1 DSM in the first phase.

	ΔV_{total} (km/s)	TOF (days)	Computation time (s)
Algorithm 1	10.418507	4976	3.290600
Algorithm 2 (3-10)	10.437478	5181	690.099926
Algorithm 3 (k=150, q=0.1)	10.418488	4977	1101.635700

Table 6.28 Comparison of the optimal solution for the mission to Saturn via flyby around Jupiter including 1 DSM in the second phase.

	ΔV_{total} (km/s)	TOF (days)	Computation time (s)
Algorithm 1	10.418505	4976	6.902500
Algorithm 2 (3-10)	10.320541	6346	1069.340430
Algorithm 3 (k=90, q=0.3)	9.934420	6435	1558.845500

Table 6.29 Comparison of the optimal solution for the mission to Saturn via flyby around Jupiter including 1 DSM in the first phase and another one in the second phase.

It seems that the optimal fuel consumption solution is again a mission without DSMs. Algorithm 3 discovers a new opportunity to reach Jupiter via a multiple revolution orbit which allows to reduce the initial impulse. Only Algorithm 3 can clearly reduce the cost of the previous direct transfer, but the computation time has been conspicuously incremented due to the complexity of the problem. The optimal solution calculated by Algorithm 3, represented on table 6.30 and figure 6.16, reduces the cost of the mission by 5%.

	Departure Date	Arrival Date	GAM Date	ΔV_{ini} (km/s)	ΔV_{end} (km/s)
n=1,m=0	17/01/2024	01/09/2041	29/10/2032	6.246267	3.617622
n=0,m=1	05/11/2025	22/06/2039	06/11/2030	6.603801	3.814393
n=1,m=1	17/01/2024	01/09/2041	30/10/2032	6.253745	3.617684

ΔV_{GAM} (km/s)	$r_p/R_{\mathrm{♃}}$	ΔV_{DSM1} (km/s)	DSM_1 Date	ΔV_{DSM2} (km/s)	DSM_2 Date
0.060695	34.5	$1.593 \cdot 10^{-7}$	01/06/2028	/	/
$2.944 \cdot 10^{-4}$	10.01	/	/	$9.569 \cdot 10^{-8}$	28/02/2035
0.062990	34.5	$7.760 \cdot 10^{-8}$	01/06/2028	$6.602 \cdot 10^{-7}$	01/04/2037

Table 6.30 Optimal solution for the mission to Saturn via GAM around Jupiter adding n DSMs in the first phase and m DSMs in the second phase.

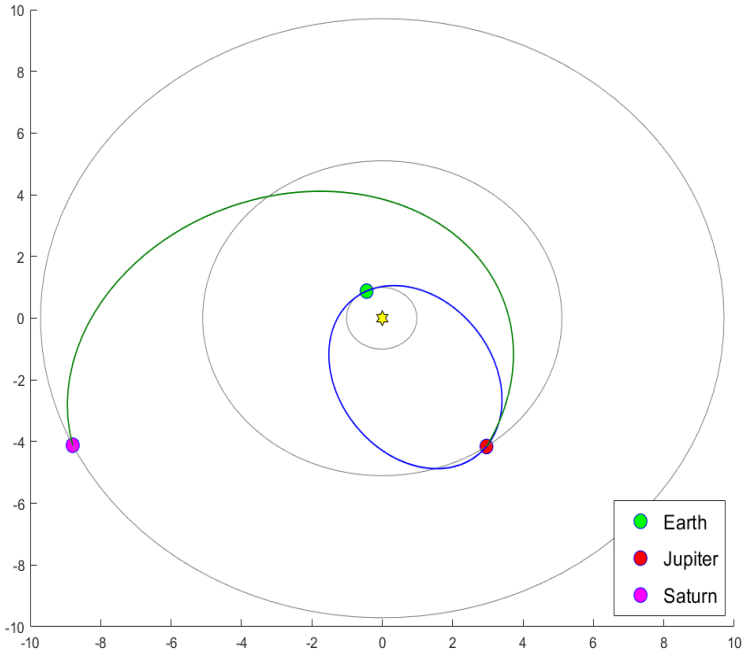

Figure 6.16 Improved optimal solution for the mission to Saturn via flyby around Jupiter.

Mars flyby

By the same token, the initial guess for the dates required by Algorithm 1 is the optimal solution without the Deep Space Maneuvers using Algorithm 3. Then, the range of the arrival time is extended until 01/01/2050 to apply Algorithms 2 and 3.

	ΔV_{total} (km/s)	TOF (days)	Computation time (s)
Algorithm 1	9.620917	2743	4.572094
Algorithm 2 (3-10)	9.114878	4424	195.977410
Algorithm 3 (k=90, q=0.3)	9.109455	4428	607.095271

Table 6.31 Comparison of the optimal solution for the mission to Saturn via flyby around Mars including 1 DSM in the first phase.

	ΔV_{total} (km/s)	TOF (days)	Computation time (s)
Algorithm 1	9.185595	2855	27.021445
Algorithm 2 (3-10)	9.317684	3157	586.400497
Algorithm 3 (k=70, q=0.3)	9.129538	3171	761.263765

Table 6.32 Comparison of the optimal solution for the mission to Saturn via flyby around Mars including 1 DSM in the second phase.

	ΔV_{total} (km/s)	TOF (days)	Computation time (s)
Algorithm 1	17.174659	2856	8.608804
Algorithm 2 (3-10)	9.099575	4424	952.199953
Algorithm 3 (k=90, q=0.3)	9.090480	4431	2263.397990

Table 6.33 Comparison of the optimal solution for the mission to Saturn via flyby around Mars including 1 DSM in the first phase and another one in the second phase.

The first aspect that stands out is the fact that it has not been found a better solution than the one associated with the mission without DSMs. As it has been proved, including intermediate DSMs becomes useful when the spacecraft travels more than one revolution between both planets under consideration because it reduces the arrival velocity, decreasing the final impulse required. Nonetheless, due to the fact that Mars is really far from Saturn, it seems that the reduction of the velocity before the gravity assist is not beneficial since the vehicle would not be able to reach the destiny planet. Moreover, as it happened in the previous case, more than one revolution in the second phase is not possible within the available interval of time, limited until 2050.
Algorithms 2 and 3 provide similar solutions while Algorithm 1 does not perform well at all on the problem with DSM in both phases. In that last case, the computation time associated with Algorithm 3 has extremely increased. The solutions calculated using Algorithm 3 are shown on table 6.34. The one associated with the lower cost does not include DSMs and it is presented in figure 6.17.

	Departure Date	Arrival Date	GAM Date	ΔV_{ini} (km/s)	ΔV_{end} (km/s)
n=1,m=0	16/11/2024	31/12/2036	13/01/2028	4.004908	4.866648
n=0,m=1	08/09/2025	15/05/2034	30/09/2027	6.603801	3.814393
n=1,m=1	13/11/2024	31/12/2036	10/01/2028	4.716479	4.407551

ΔV_{GAM} (km/s)	h_p (km)	ΔV_{DSM1} (km/s)	DSM_1 Date	ΔV_{DSM2} (km/s)	DSM_2 Date
0.237906	1310.9958	$2.445 \cdot 10^{-7}$	03/07/2026	/	/
0	708.4038	/	/	$5.508 \cdot 10^{-3}$	12/05/2030
0.218067	1423.3598	$4.047 \cdot 10^{-5}$	05/07/2026	0	01/07/2032

Table 6.34 Optimal solution for the mission to Saturn via GAM around Mars adding n DSMs in the first phase and m DSMs in the second phase.

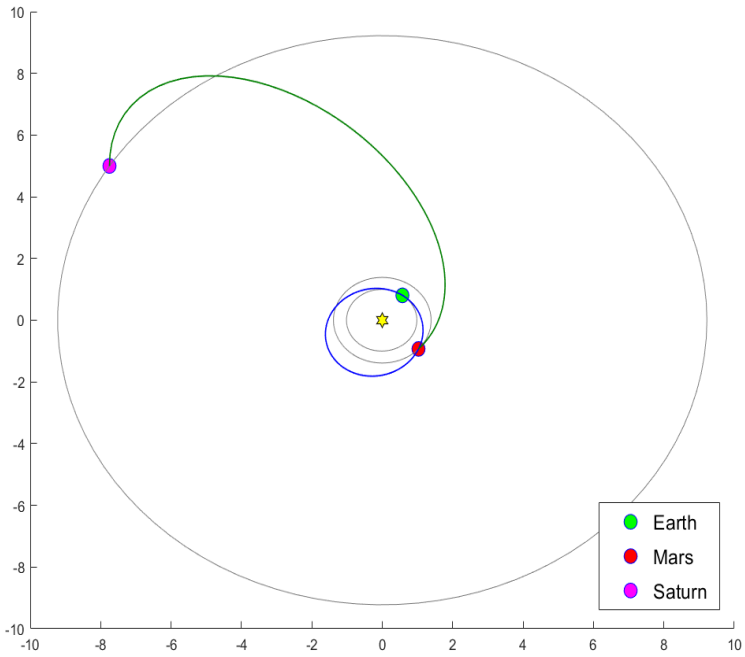

Figure 6.17 Optimal solution for the mission to Saturn via fly by Mars with multiple revolutions in the first phase.

6.3 Mission to Jupiter

Eventually, the analysis is focused on a mission to Jupiter. Jupiter is, by far, the largest planet in the Solar System and it is surrounded by dozens of moons. Pioneer 10 was the first spacecraft to visit the Gas Giant, when it flew by Jupiter in 1973. Recently, in 2016, NASA's mission Juno entered an orbit around Jupiter in order to study its formation and evolution, as well as to explore the planet's deep atmosphere. The mission evaluated in this section departs from a parking orbit at an altitude $h_{park1} = 200$ km over the Earth and arrives at another parking orbit at an altitude $h_{park2} = 9R_{♃}$ over Jupiter. It is important to remind that the symbol ♃ refers to the planet Jupiter.

6.3.1 Direct Mission

First, the direct transfer is evaluated. As there are no references to set a range of possible initial dates for this mission, there has been carried out an investigation to find the optimal departure date. Initial dates from 01/01/2021 to 01/01/2026 have been tested and it has been found an optimal solution on the second half of 2025. Since the Earth-Jupiter synodic period is $T_{syn} = 399$ days, the range of the departure time T_{ini} is lying between 01/03/2025 and 01/09/2026.

To estimate the necessary time of flight to go from the Earth to Jupiter, the time of flight associated with the Hohmann transfer is calculated (about 998 days, approximately 3 years). Then, the range of the arrival date is established between 01/01/2027 and 01/01/2030.

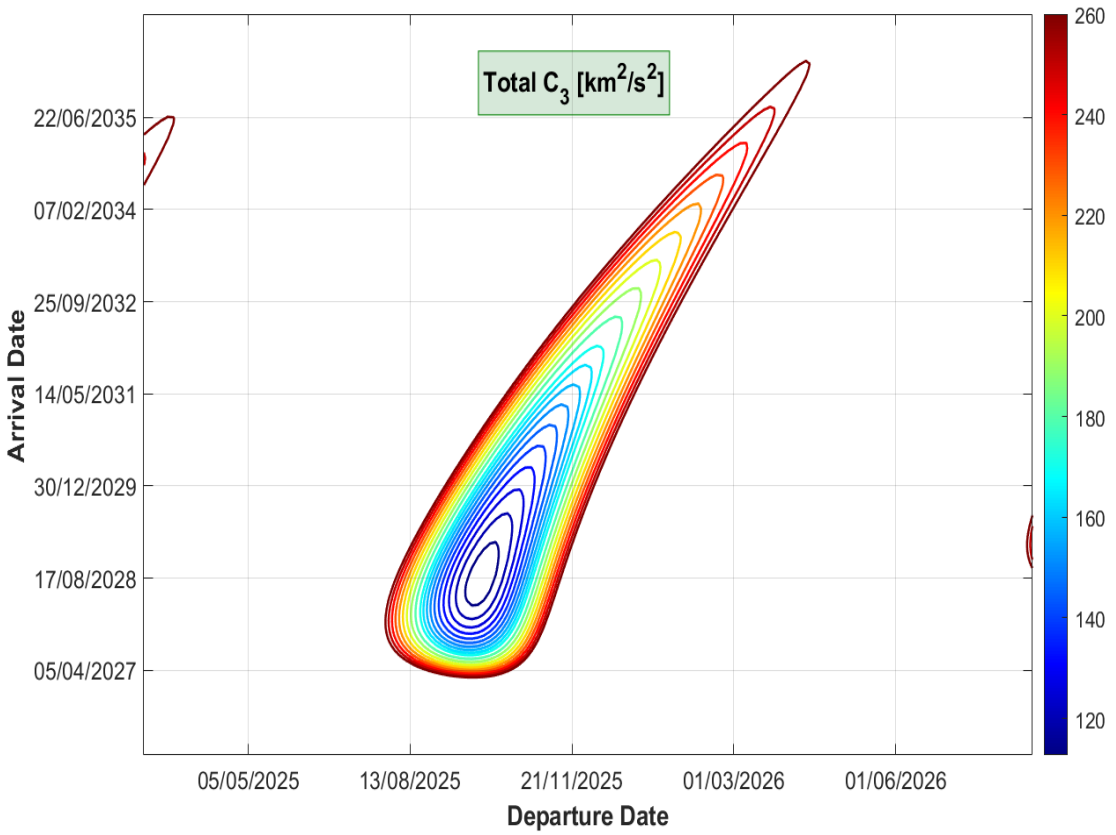

Figure 6.18 Pork chop plot: Direct mission to Jupiter.

After having built the pork-chop plot associated with this mission (figure 6.18), an optimal solution is detected. It involves a departure date around the month of October of 2025 and an arrival date around August

of 2028.

Then, *fmincon* is employed to search the optimal point shown in the pork-chop plot, operating with the initial guess ($T_{ini} = 01/10/2025$, $T_{end} = 01/08/2028$). The solver successfully finds the optimal point shown in figure 6.4, which is detailed on table 6.35 and figure 6.19:

	Departure Date	Arrival Date	ΔV_{total} (km/s)	ΔV_{ini} (km/s)	ΔV_{end}(km/s)
Direct Mission	25/09/2025	16/08/2028	12.664042	6.406980	6.257062

TOF (days)	Computation time (s)
1056	2.486128

Table 6.35 Optimal solution for the direct mission to Jupiter.

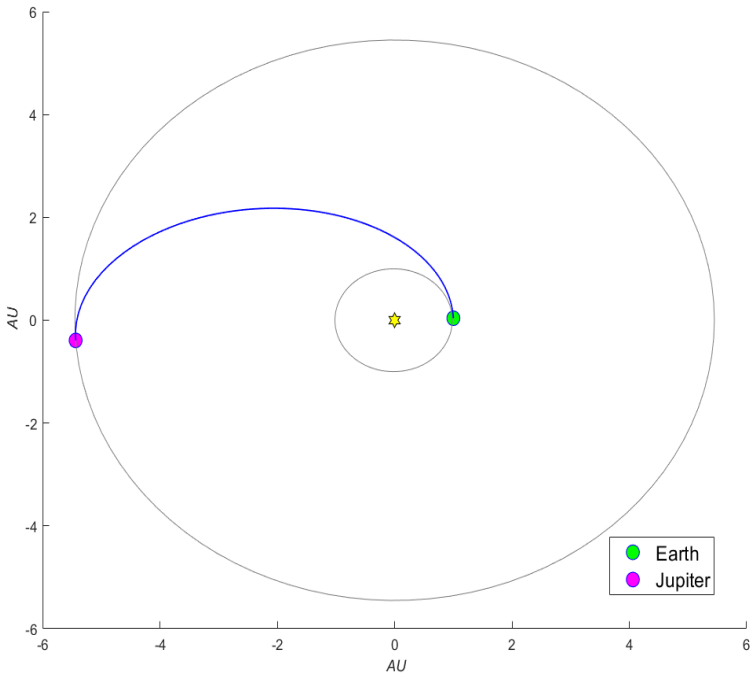

Figure 6.19 Optimal solution for the direct mission to Jupiter.

The optimal solution found is similar to a Hohmann transfer and the associated cost seems to be very expensive, thus it must find an alternative way to reach Jupiter.

6.3.2 Gravity Assist Maneuvers

The cost of the direct mission is considerably high hence, in view of the results of the previous section, it was decided to include Gravity Assist Maneuvers in the mission. It was expected that the time of flight would also be increased, yet it would not be a serious problem. The range of the initial date is broadly establish as (01/06/2024, 01/06/2026).

Mars flyby

Firstly, a swing by Mars is evaluated. In Section 6.2, an optimal solution for a direct Earth-Mars transfer was found in May 2025, where the arrival date at Mars takes place in July 2027. That solution is used as the initial guess for Algorithm 1, calculating the final date adding the time of flight of the Hohmann transfer of Mars-Jupiter (3 years). The lower bound associated with the final date is set as 01/01/2028, and its upper bound is 01/01/2032.

Algorithms 2 and 3 find similar solutions, outperforming Algorithm 1. The optimal solution found using Algorithm 3 reduces the cost of the direct mission by 21.5%, and it is presented on table 6.37 and figure 6.20

	ΔV_{total} (km/s)	TOF (days)	Computation time (s)
Algorithm 1	10.612632	2080	1.969129
Algorithm 2 (3-5)	9.943996	1694	114.962267
Algorithm 3 (k=150, q=0.3)	9.938240	1654	406.142325

Table 6.36 Comparison of the optimal solution for the mission to Jupiter via GAM around Mars obtained using each algorithm.

Departure Date	Arrival Date	GAM Date	ΔV_{ini} (km/s)	ΔV_{end} (km/s)	ΔV_{GAM} (km/s)
08/01/2025	20/07/2029	02/05/2026	3.896029	6.042191	$1.938 \cdot 10^{-5}$

h_{mars} (km)	δ (º)
285.7283	15.482

Table 6.37 Optimal solution for the mission to Jupiter via GAM around Mars obtained using Algorithm 3.

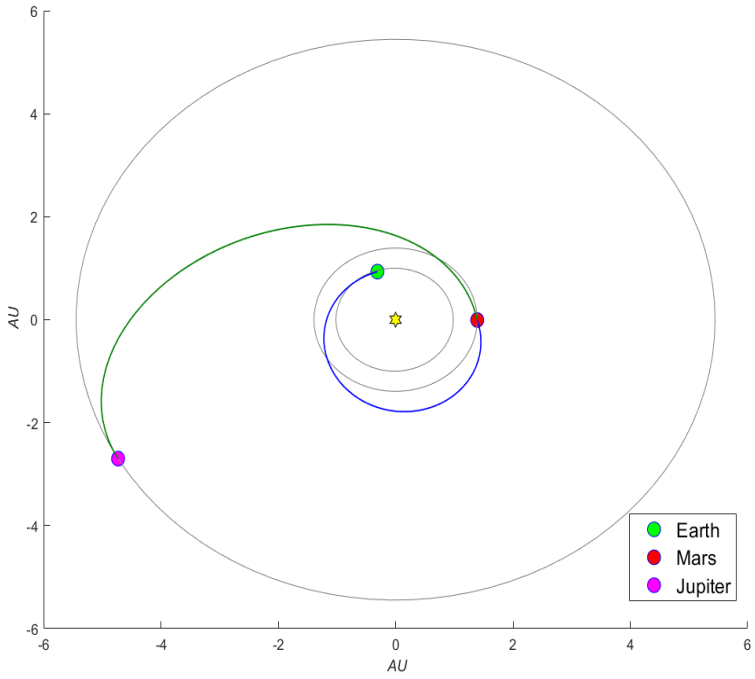

Figure 6.20 Optimal solution for the mission to Jupiter via GAM around Mars using Algorithm 3.

Once again, the use of a Gravity Assist Maneuver allows to reduce the total cost of the mission. In this case, the initial impulse is strongly decreased due to the proximity of Mars to the Earth. However, the final impulse is still high.

Venus flyby + Mars flyby

Secondly, a GAM around Venus is added before the Mars flyby. Having discovered an optimal solution on January 2025 for the direct mission from the Earth to Venus, it is used as the initial guess for the departure date. Then, obtaining the respective times of flight associated with each Hohmann trajectory, the initial guess for the rest of the dates is determined. Since the estimated total time of flight, calculated using Hohmann tranfers, is about 4 years, the range of the arrival dates lies between 01/01/2028 and 01/01/2033.

Algorithms 2 and 3 perform better than Algorithm 1, although they need a large computation time, especially Algorithm 3, since 200 ants are needed to find that solution. The solution discovered by Algorithm 3, represented on table 6.39 and figure 6.21, reduces the total cost of the direct mission by 17%, adding 500 days to the total time of flight.

	ΔV_{total} (km/s)	TOF (days)	Computation time (s)
Algorithm 1	11.992899	1433	6.473917
Algorithm 2 (5-10)	10.581271	2159	522.631001
Algorithm 3 (k=200, q=0.3)	10.455651	1547	2181.518370

Table 6.38 Comparison of the optimal solution for the mission to Jupiter via 2 GAMs (around Venus and Mars) obtained using each algorithm.

Departure Date	Arrival Date	$GAM_{♀}$ Date	$GAM_{♂}$ Date	ΔV_{ini} (km/s)	ΔV_{end} (km/s)
03/04/2025	29/06/2029	26/09/2025	22/05/2026	4.430988	6.018253

$\Delta V_{GAM_{♀}}$ (km/s)	$\Delta V_{GAM_{♂}}$ (km/s)	$h_{p♀}$ (km)	$h_{p♂}$ (km)	$\delta_{♀}$ (º)	$\delta_{♂}$ (º)
$1.048 \cdot 10^{-3}$	$5.361 \cdot 10^{-3}$	363.7370	286.9712	32.630	20.315

Table 6.39 Optimal solution for the mission to Jupiter via 2 GAMs (around Venus and Mars) calculated using Algorithm 3.

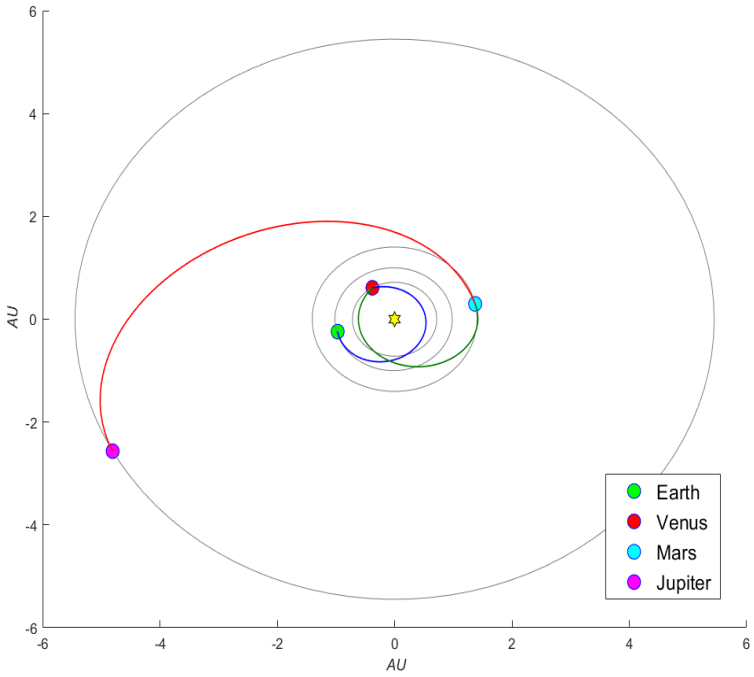

Figure 6.21 Optimal solution for the mission to Jupiter via 2 GAMs (around Venus and around Mars) calculated using Algorithm 3.

Venus flyby + Earth flyby

Next, the mission with a first GAM around Venus and a second one around the Earth is analysed. The same initial guess as in the previous case is used for the initial and the Venus flyby dates. Then, the times of flight of the respective direct transfers are added. The departure date's range is set as (01/01/2028, 01/01/2032).

	ΔV_{total} (km/s)	TOF (days)	Computation time (s)
Algorithm 1	11.982431	1829	3.447780
Algorithm 2 (3-8)	10.328592	1770	326.222310
Algorithm 3 (k=200, q=0.3)	10.255098	1608	1542.5592

Table 6.40 Comparison of the optimal solution for the mission to Jupiter via 2 GAMs (around Venus and the Earth) obtained using each algorithm.

This time, the computation time required by Algorithm 3 to improve the solution found by Algorithm 2 is almost 5 times higher, since 200 ants are needed. However, that solution, presented on table 6.41 and figure 6.22, achieves a total cost 19% lower than the direct transfer, considerably reducing the initial impulse.

Departure Date	Arrival Date	$GAM_{♂}$ Date	$GAM_{⊕}$ Date	ΔV_{ini} (km/s)	ΔV_{end} (km/s)
07/03/2025	01/08/2029	20/08/2025	22/09/2026	3.933149	6.316399

$\Delta V_{GAM_{♂}}$ (km/s)	$\Delta V_{GAM_{⊕}}$ (km/s)	$h_{p♂}$ (km)	$h_{p⊕}$ (km)	$\delta_{♂}$ (º)	$\delta_{⊕}$ (º)
$3.178 \cdot 10^{-3}$	$2.372 \cdot 10^{-3}$	636.7420	1017.0090	50.278	33.759

Table 6.41 Optimal solution for the mission to Jupiter via 2 GAMs (around Venus and around the Earth) calculated using Algorithm 3.

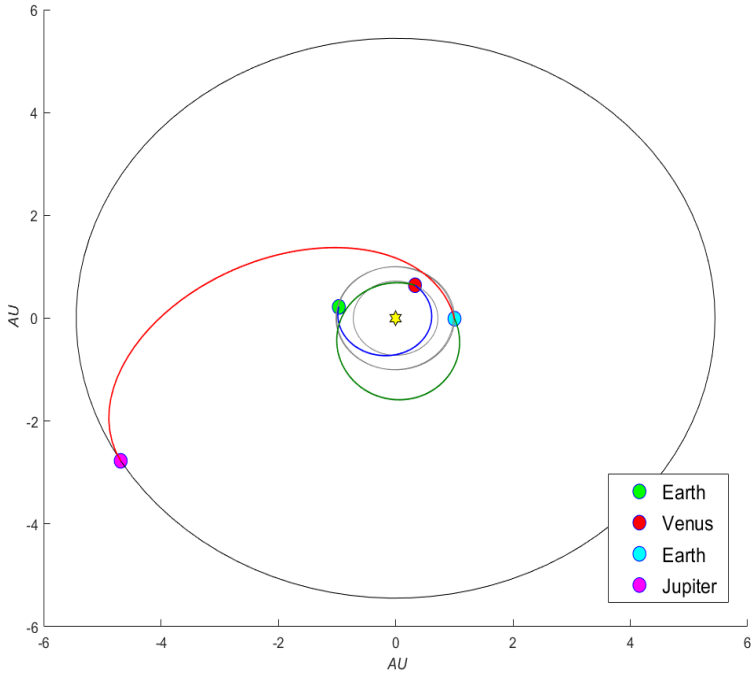

Figure 6.22 Optimal solution for the mission to Jupiter via 2 GAMs (around Venus and around the Earth) calculated using Algorithm 3.

Mars flyby + Earth flyby

Subsequently, the mission with a swing by Mars and another one by the Earth is analysed. The optimal solution previously found for the mission with a GAM around Mars is employed to set the initial and the Mars flyby dates. The rest of the dates are calculated adding the times of flight of the Hohmann transfer of each phase. The range of possible arrival dates is established as (01/01/2028,01/01/2035), since the total estimated Hohmann's time of flight is slightly superior to 4 years.

	ΔV_{total} (km/s)	TOF (days)	Computation time (s)
Algorithm 1	13.125564	2341	2.849000
Algorithm 2 (3-8)	11.869616	1994	203.541980
Algorithm 3 (k=400, q=0.3)	10.649167	1443	5485.7480

Table 6.42 Comparison of the optimal solution for the mission to Jupiter via 2 GAMs (around Mars and the Earth) obtained using each algorithm.

Finding a great optimal solution was hard for Algorithm 3, with a computational time which went through the roof. This time, Algorithm 1 could not obtain a solution which enhances the one associated with the

direct mission. The solution achieved by Algorithm 3, shown on table 6.43 and figure 6.23, lessens the total cost of the direct mission by 16%, increasing the time of flight by 400 days.

Departure Date	Arrival Date	$GAM_{♂}$ Date	$GAM_{⊕}$ Date	ΔV_{ini} (km/s)	ΔV_{end} (km/s)
07/10/2024	19/09/2028	30/03/2025	18/11/2025	4.217310	6.428591

$\Delta V_{GAM_{♂}}$ (km/s)	$\Delta V_{GAM_{⊕}}$ (km/s)	$h_{p♂}$ (km)	$h_{p⊕}$ (km)	$\delta_{♂}$ (º)	$\delta_{⊕}$ (º)
$1.701 \cdot 10^{-4}$	$6.696 \cdot 10^{-3}$	286.9468	1051.7345	21.612	20.177

Table 6.43 Optimal solution for the mission to Jupiter via 2 GAMs (around Mars and around the Earth) calculated using Algorithm 3.

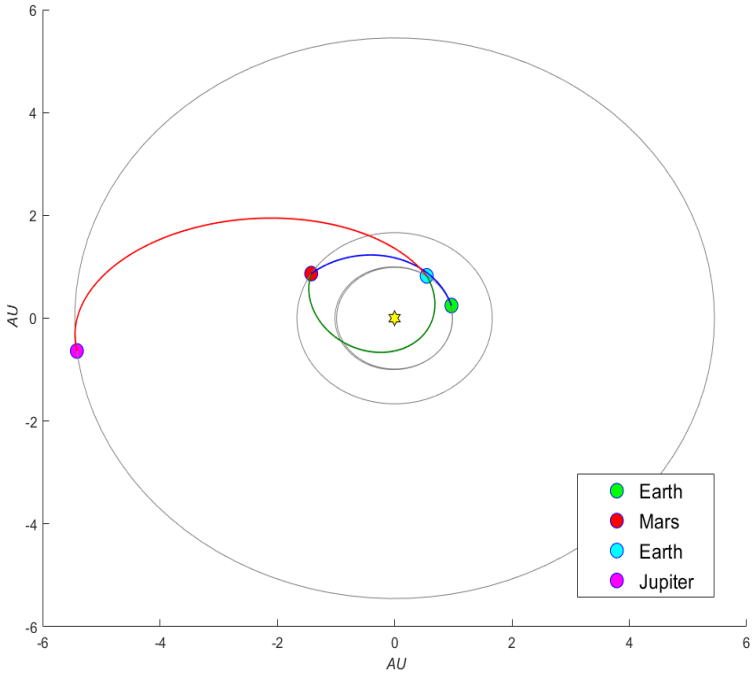

Figure 6.23 Optimal solution for the mission to Jupiter via 2 GAMs (around Mars and around the Earth) calculated using Algorithm 3.

Venus flyby + Earth flyby + Mars flyby

Lastly, a mission which involves three flybys (Venus, the Earth and Mars) is analysed. The solution for the direct mission to Venus is employed as the initial guess, adding the required times of flight until reaching Jupiter. The arrival date is lying between 01/01/2028 and 01/01/2033.

	ΔV_{total} (km/s)	TOF (days)	Computation time (s)
Algorithm 1	17.544543	1624	8.036202
Algorithm 2 (3-8)	15.995125	4839	468.122610
Algorithm 3 (k=200, q=0.3)	11.534919	1733	1745.478800

Table 6.44 Comparison of the optimal solution for the mission to Jupiter via 3 GAMs (around Venus, the Earth and Mars) obtained using each algorithm.

Only Algorithm 3 is able to reach a solution with a lower total cost than the direct mission to Jupiter. Although that solution reduces the cost by 9%, it requires a high initial impulse, as it is presented on table 6.45 and figure 6.24.

Departure Date	Arrival Date	GAM$_{\varphi}$ Date	GAM$_{\oplus}$ Date	GAM$_{\sigma}$ Date	ΔV_{ini} (km/s)
30/06/2024	30/03/2029	18/02/2025	06/05/2025	10/10/2026	5.374171

ΔV_{end} (km/s)	$\Delta V_{GAM_{\varphi}}$ (km/s)	$\Delta V_{GAM_{\oplus}}$ (km/s)	$\Delta V_{GAM_{\sigma}}$ (km/s)	$h_{p\varphi}$ (km)	$h_{p\oplus}$ (km)
6.157258	$1.235 \cdot 10^{-4}$	$1.719 \cdot 10^{-3}$	$1.648 \cdot 10^{-3}$	370.7752	1085.9856

$h_{p\sigma}$ (km)	δ_{φ} (º)	$\delta_{\oplus}$ (º)	δ_{σ} (º)
300.8328	78.679	81.578	11.447

Table 6.45 Optimal solution for the mission to Jupiter via 3 GAMs (around Venus, the Earth and Mars) calculated using Algorithm 3.

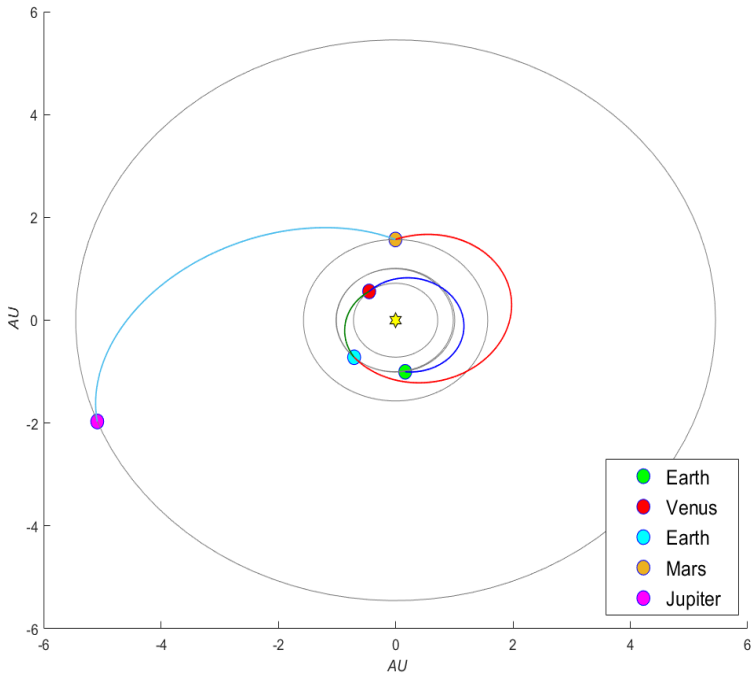

Figure 6.24 Optimal solution for the mission to Jupiter via 3 GAMs (around Venus, the Earth and Mars) calculated using Algorithm 3.

Final comparison

In conclusion, the results reached by each algorithm are included in a table, in order to compare them.

	ΔV_{total} (km/s)	TOF (days)	ΔV_{ini} (km/s)	ΔV_{end} (km/s)
Direct	12.6640	1056	6.4070	6.2571
Mars flyby	10.6126	2080	.2997	6.3126
Venus + Mars	11.9929	1433	4.0185	6.4396
Venus + Earth	11.9824	1829	4.0119	6.6568
Mars + Earth	13.1256	2341	4.4825	6.9572
Venus + Earth + Mars	17.5445	1624	6.4084	6.4303

Table 6.46 Optimal solution for every GAM mission to Jupiter using Algorithm 1.

	ΔV_{total} (km/s)	TOF (days)	ΔV_{ini} (km/s)	ΔV_{end} (km/s)
Direct	12.6640	1056	6.4070	6.2571
Mars flyby	9.9440	1694	3.9002	6.0436
Venus + Mars	10.5813	2159	4.1637	6.4132
Venus + Earth	10.3276	1770	3.9656	6.3559
Mars + Earth	11.8696	1994	5.3856	6.4828
Venus + Earth + Mars	15.9951	4839	6.9807	8.8102

Table 6.47 Optimal solution for every GAM mission to Jupiter using Algorithm 2.

	ΔV_{total} (km/s)	TOF (days)	ΔV_{ini} (km/s)	ΔV_{end} (km/s)
Direct	12.6640	1056	6.4070	6.2571
Mars flyby	9.9382	1654	3.8960	6.0422
Venus + Mars	10.4557	1547	4.4310	6.0183
Venus + Earth	10.2551	1608	3.9331	6.3164
Mars + Earth	10.6998	1443	4.2173	6.4286
Venus + Earth + Mars	11.5349	1733	5.3742	6.1573

Table 6.48 Optimal solution for every GAM mission to Jupiter using Algorithm 3.

In brief, every algorithm is able to calculate a solution with a lower cost than the direct mission. Nevertheless, Algorithm 3 and Algorithm 2 improve the direct mission for every case under consideration except for the 3 GAMs problem. The major obstacle of the mission to Jupiter is the final impulse, which has not been reduced enough in any of the missions. That point is the main difference between this case and the mission to Saturn, which allowed to reduce the cost of the direct mission by 35% using Gravity Assist Maneuvers. In light of the results shown on table 6.48, the optimal solution to reach Jupiter seems to be a fly by Mars, which reduces the cost by 21.5% adding 638 days to the direct mission. Summarising, a mission to Saturn is cheaper than a mission to Jupiter, yet the time of flight required for the first one is extremely higher.

7 Conclusions and Future Work

7.1 Conclusions

This project has presented a novel approach to the preliminary design of interplanetary trajectories. The aim of this study was to explore smart ways of using local-optimization based methods in order to perform a global minimum search. This work has demonstrated the efficiency of such approach both in terms of local optimal solution improvement as well as maintaining a reasonable low computational burden (which is usually not the case for Metaheuristic algorithms).

Three algorithms of various complexity have been tested. Algorithm 1 yielded significant speed advances when the problem under consideration was not excessively complicated. Nonetheless, it was actually dependent on the previous analysis. Algorithm 2 has performed pretty well, providing some interesting solutions and allowing to discover unexpected ones although it requires a longer computation time. It should be noted that the use of a multi-start search employing local optimization processes which start from initial guesses randomly distributed over the search space, made it possible to identify new optimal solutions. Finally, the most important advantage of Algorithm 3 is that it can enhance the solutions which were found by Algorithm 2, adding a back-up procedure which allows to focus on the best candidates. Definitely, the Ant-Colony optimization procedure allows to achieve better solutions, deeply exploring the search space and concentrating on the most auspicious contenders. Summarising, the main achievement of the proposed algorithms is the fact that they achieved a significantly beneficial solution which appears to be a global minimum, although a local optimization procedure is employed. Moreover, the use of a local optimal searcher led to a noticeable low computation time, which is less than 30 minutes in the worst case.

Four mission analysis classes have been investigated: the two impulse direct planet-to-planet transfer problem, the Deep Space Maneuver problem, the Multiple Gravity Assist interplanetary transfer problem and the problem which involves a Gravity Assist and Deep Space Maneuvers. By considering the objective function value reached at the end of the optimization process and the required computation time to get it, the results of the test phase can be summarised as:

- Direct transfer: The optimal solution found is always a kind of Hohmann transfer when trajectories with complete revolutions around the Sun are not evaluated. The pork chop plot shows one clear optimal point in terms of fuel consumption, due to the fact that the 2-D problem is considered. The required computation time is very short, and every algorithm can discover the same solution.
- DSM problem: It has been proved in the mission to Mars that a DSM can reduce the necessary total cost. Generally, they help to reduce the final impulse and they increase the required total time of flight indeed. Moreover, the investigation of this problem allows to discover new optimal solutions for the direct one. Since this method splits the trajectory, it is possible to find optimal transfers with complete revolutions around the Sun where the DSM are zero which enhance the previously discovered optimal direct trajectory.
- MGA problem: The results demonstrate that the proposed methodology can effectively tackle the complexity of multiple Gravity Assist interplanetary transfers while avoiding being trapped in local minima. Furthermore, it was proved that Gravity Assist Maneuvers permit to substantially decrease the

total cost of the mission. Particularly, Algorithm 1 did not perform very well when analysing multiple maneuvers since the initial guesses for the transfer times between planets were not accurate.

- GAM + DSM: For the studied case, the total cost is reduced by finding a solution with a complete revolution around the Sun with no intermediate impulses.

The proposed methodology has been tested on a suite of typical mission design problems of high complexity and it has been proved that favourable solutions can be obtained. DSMs were very useful in the mission to Mars while GAMs became essential in the missions to Saturn and Jupiter to reduce the associated total cost.

7.2 Future Work

In view of the calculated results, in order to improve the them and to overcome potential problems, recommendations for future work are listed as follows:

- Since it has been proved that multiple revolution trajectories frequently reduce the total cost of a direct transfer, future research should employ a Lambert's Problem solver with the possibility of multiple revolution orbits.
- The upper limitation of the arrival date affected the achieved results, thus it would be interesting to use Ephemeris with a major valid range in order to discover new solutions. Moreover, future studies might consider not only planetary Ephemeris but also other bodies' such as comets and asteroids, to analyse missions like Rosetta.
- It is also recommended to investigate the launching phase, since it would provide a more realistic idea of the mission requirements.
- Future investigations could explore low thrust maneuvers, instead of performing Deep Space Maneuvers, studying the electric propulsion feasibility.
- Finally, the 3-D problem should be considered in order to design interplanetary transfers which would be closer to reality.

List of Figures

List of Tables

Bibliography

[1] Voyager, The Interstellar Mission, extracted on may 2020 from *http://thehigherlearning.com/2013/12/03/voyager-mission/*

[2] Howard Curtis, *Orbital Mechanics for Engineering Students*, Elsevier, 2005.

[3] Alberto Lobato, *Análisis y Diseño de una Misión Interplanetaria a un Planeta Exterior*, Seville University, 2012.

[4] Krzysztof Socha and Marco Dorigo, *Ant Colony Optimization for Continuous Domains*, Elsevier,2006.

[5] Shahrbanou Madadgar and Abbas Afshar, *An Improved Continuous Ant Algorithm for Optimization of Water Resources Problems*, Water Resources Management, vol. 23, 2119-2139, 2014.

[6] Matthew A. Vavrina, Jacob A. Englander and Donald H. Ellison, *Global Optimization of N-Maneuver, High-Thrust Trajectories using Direct Multiple Shooting*, Conference Paper, 2016.

[7] E. M. Standish, *Keplerian Elements for Approximate Positions of the Major Planets*, Solar System Dynamics Group, JPL / Caltech.

[8] Rafael Vazquez Valenzuela, *Orbital Mechanics and Spacecrafts*, Class notes (ETSI, Seville), extracted from *http://aero.us.es/astro/desc.html*.

[9] Giancarlo Genta, *Next Stop Mars: The Why, How and When of Human Missions*, Springer, 324, 2017.

[10] Sergey Zaborsky, *Analytical Solution of Two-Impulse Transfer Between Coplanar Elliptical Orbits*, Journal of Guidance Control and Dynamics, vol. 37, 2014.

[11] Joris T. Olympio and Jean-Paul Marmorat, *Global Trajectory Optimisation: Can we prune the solution space when considering Deep Space Maneuvers?*, Final Report, 2008.

[12] Pierluigi Di Lizia and Gianmarco Radice, *Advanced Global Optimisation: Tools for Mission Analysis and Design* University of Glasgow, 2004.

[13] Massimiliano Vasile and Paolo DePascale, *On the Preliminary Design of Multiple Gravity-Assist Trajectories*, Politecnico of Milano/University of Padova, 2011.

[14] Khurram Qadir, *Multi-gravity Assist Design Tool for Interplanetary Trajectory Optimisation*, Cranfield University, 2008.

[15] Guadalupe M. Vizcaíno, *Automatización del Proceso de Diseño de un Circuito Electrónico con la Herramienta MATLAB y el Simulador de Circuitos SPECTRE*, Seville University, Chapter 5, 2006.

[16] Donald Hamilton Ellison, *Robust Preliminary Design for Multiple Gravity Assist Spacecraft Trajectories*, University of Illinois, 2018.

[17] Abolfazl Shirazi, Josu Ceberio and Jose A. Lozano, *Spacecraft trajectory optimization: A review of models, objectives, approaches and solutions*, Progress in Aerospace Sciences, 2018.

[18] R.A.N. Araujo, O.C. Winter, A.F.B.A. Prado and R. Vieira Martins, *Sphere of Influence and Gravitational Capture Radius: A Dynamical Approach*, Monthly Notices of the Royal Astronomical Society, vol. 391, 675-684, 2008.

[19] Sam Wagner, Bong Wie and Brian Kaplinger, *Computational Solutions to Lambert's Problem on Modern Graphics Processing Units*, Journal of Guidance Control and Dynamics, vol. 38, 1305-1310, 2015.