CRANFIELD UNIVERSITY

A. BELLOME

TRAJECTORY DESIGN OF MULTI-TARGET MISSIONS VIA GRAPH TRANSCRIPTION AND DYNAMIC PROGRAMMING

SCHOOL OF AEROSPACE, TRANSPORT AND MANUFACTURING
Aerospace Engineering

PhD
Academic Year: 2019–2022

Supervisors: J. P. Sánchez, L. Felicetti, S. Kemble
December 2022

CRANFIELD UNIVERSITY

SCHOOL OF AEROSPACE, TRANSPORT AND MANUFACTURING
Aerospace Engineering

PhD

Academic Year: 2019–2022

A. BELLOME

Trajectory Design of Multi-Target Missions via Graph Transcription and Dynamic Programming

Supervisors: J. P. Sánchez, L. Felicetti, S. Kemble
December 2022

Abstract

Missions that can visit multiple orbital targets represent the next cornerstone for space travels, be it for science, exploration or even exploitation. The trajectory design of such missions requires to solve a mixed-integer programming problem, on which the selection of a proper sequence of targets depends upon the quality of the trajectory that links them, where quality usually refers to propellant consumption or mission duration.

Two aspects are important when addressing these problems. The first one is to identify optimal solutions with respect to critical mission parameters. Current approaches to solve these problems require computing time that rises with the number of control parameters, as the visiting objects sequence length, as well as rely on a-priori knowledge to define a manageable design space (i.e., departing dates, presence of deep space manoeuvres, etc.). Moreover, the more challenging multi-objective optimization needs to be tackled to appropriately inform the mission design with full extent of launch opportunities. The second aspect is that beyond the obvious complexity of such problems formulation, preliminary mission design requires not only to locate the global optimum solutions but, also, to map the ensemble of solutions that leads to feasible transfers.

This thesis describes a pipeline to transcribe the mixed-integer space into a discrete graph made by grids of interconnected nodes for missions that visit multiple celestial objects, like planets, asteroids, comets, or a combination thereof, by means of one single spacecraft. This allows to exploit optimal substructure of such problems, opening dynamic programming to be conveniently applied. Dynamic programming principles are thus extended to multi-objective optimization of such trajectories and used to explore the transcribed graph, guaranteeing Pareto optimality with efficient computational effort. A modified dynamic programming approach is also derived that allows to retain more and diverse solutions in the final set compared to known standard approaches, while guaranteeing global optimality on the transcribed space.

Numerous applications are presented where such pipeline is successfully applied. Trajectories towards Jupiter and Saturn alongside novel transfers for comet sample return missions are discussed, as well as trajectories that visit multiple asteroids in the main belt. Such scenarios prove robustness and efficiency of proposed approaches in capturing optimal solutions and wide Pareto fronts on search spaces of complex configuration.

Keywords

Mission analysis; Combinatorial optimization; Dynamic programming; Multiple Gravity Assist; Main Asteroids Belt.

Contents

List of Figures

List of Tables

List of Abbreviations

ACO	Ant Colony Optimization
AU	Astronomical Unit
BB	Branch and Bound
BF	Breadth First
BS	Beam Search
BW	Beam Width
COE	Classical Orbital Element
CR3BP	Circular Restricted Three Body Problem
CSP	Constraint Satisfaction Problem
DF	Depth First
DSM	Deep Space Manoeuvre
E	Earth
ESA	European Space Agency
GTOC	Global Trajectory Optimization Competition
GVE	Gauss Variational Equation
HOCP	Hybrid Optimal Control Problem

J	Jupiter
M	Mars
MAB	Main Asteroid Belt
MGA	Multiple Gravity Assist
MINLP	Mixed-Integer Non-Linear Programming
MODP	Multi-Objective Dynamic Programming
MOID	Minimum Orbital Interception Distance
NASA	National Aeronautics and Space Administration
PSO	Particle Swarm Optimization
S	Saturn
SODP	Single-Objective Dynamic Programming
SOI	Sphere of Influence
STM	State Transition Matrix
TPBVP	Two-Point Boundary Value Problem
TSP	Traveling Salesman Problem
V	Venus
VILT	V-Infinity Leveraging Transfer
Y	Mercury

Acknowledgments

I would like to thank my supervisory team, for which I feel blessed. Firstly, I would like to thank Dr. Joan-Pau Sánchez for the dedication, attention, vision and support he provided throughout these three years. He has been helpful and flexible, always ready to provide both technical and personal advice. Likewise, I would like to sincerely thank Stephen Kemble for his experience and knowledge. He has always been passionate about our work, and he has never spared advice and suggestions without which this work would probably not have succeeded. Last but not least, I would like to thank Dr Leonard Felicetti, who turned out to be more like an elder brother, before being a supervisor. I will never forget the Sunday meetings, the tears and joys. Thank you for putting up with me and supporting me. I will always carry with me the friendship that binds us.

I would also like to use these lines to thank my family, my mother, father and sister, for always being there for me, and for never letting me lack support and love. I also take this opportunity to thank my friends, who made the Ph.D. experience so much more pleasant and enjoyable. In particular, I thank Filippo Z., for our chats, Giovanni D., for being my friend for 20 years, and Lorenzo P., because I always feel you by my side. I also thank my *D&D* group... screw the Ph.D. ... let's roll!

Finally, I can only thank my other family, for which I am and will be eternally grateful. Thanks to Sam, for being the best, sweetest, furriest dog in the world. And of course, thank you Elena for always being by my side and for the support you never let me lack. These years have flown with you, and the ones ahead will be even better.

Chapter 1

Introduction

In the past, designing spacecraft trajectories was limited to the definition of transfers to single destinations in space. In the early decades of space exploration, spacecraft usually targeted single objects, from the Moon [1] to comets [2], to Solar System planets, such as Venus [3], Mars [4] or even Jupiter [5]. New technologies, and a more daring approach to space, present nowadays the challenge of missions that target multiple destinations, i.e., the so-called multi-target missions, be it for science, exploration or even exploitation.

Instances of mission design problems that target multiple orbital way-points, instead of one single target [1], represent a growing trend. Some example of this include, but are not limited to: (1) ESA's JUICE mission [6], which will perform more than twenty close passages with Jovian moons; (2) commercial concepts for On-orbit Servicing [7], among which Active Debris Removal [8] is one of the most famous, considering to rendezvous with multiple objects by means of one single platform to help restoring Earth orbital environment; (3) missions that aim to pass by multiple asteroids in the main asteroid belt (MAB), as proposed by CASTAway [9,10] in the context of ESA's Medium Class mission call, or as Lucy mission [11], that aims to pass-by 1 MAB object and 7 Trojan asteroids.

[1] In this thesis, the words targets and way-points are used interchangeably to refer to orbital positions that a probe should visit along its mission.

Figure 1.1 represents an example trajectory for a multi-target exploration of the MAB as proposed for CASTAway mission. In this example, a spacecraft visits 12 orbital waypoints, that are a combination of two planetary encounters, one with Mars and one with the Earth, and close passages with ten asteroids. The overall strategy assumes a launch from the Earth on the 21st of January 2037 and the help of Mars and Earth gravity to increase spacecraft apoapsis for in-depth study of different objects in the MAB.

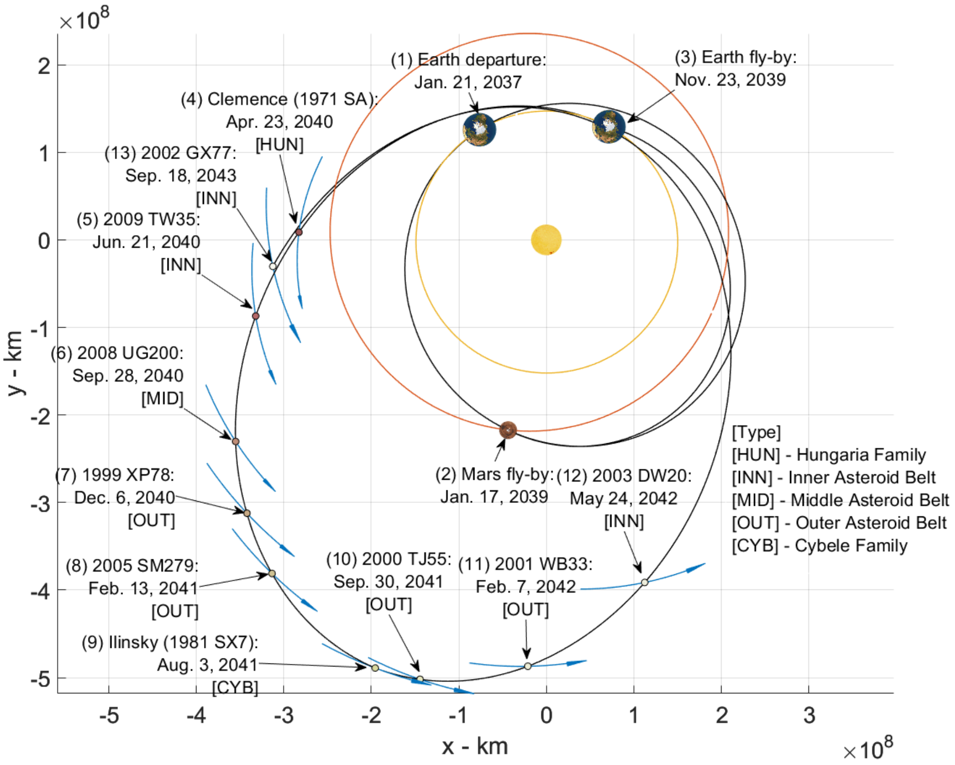


Figure 1.1: Example of multi-target exploration of the MAB by means of planetary fly-bys at Mars and Earth. Arrowed lines represent asteroids paths. Sun, planets and asteroids are not in scale.

The present thesis deals with the mission analysis part of an overall space mission life cycle, which is one of the first steps on the definition of the space mission itself. Particularly, this work addresses challenges and approaches associated to analysis of space missions that aim to visit multiple orbital targets by means of one single spacecraft, aspiring to provide design strategies that are efficient, scalable, and robust for addressing preliminary steps of such space missions.

This Chapter serves as presentation of the overall work and introduces the context of the thesis. It is divided as follows: section 1.1 describes the main steps of current mission design phases, presenting the overall mission operations life cycle, with an eye on the preliminary definition of scientific goals and assessment of potential trajectories on the preliminary phase of the design; section 1.2 defines the role of the mission analysis and discusses its main ingredients; section 1.3 discusses the scientific potential and current interests around multi-target missions; section 1.4, discusses the purposes of the present thesis, highlighting its aim and objectives; section 1.5 finally presents the structure of the present thesis alongside main research contributions.

1.1 Space Mission Design Overview

Space missions can be seen as the sum of so-called mission operations [12], which are all the activities that define the way for a spacecraft to be designed, built and to accomplish its scientific objectives. Particularly, mission operations include different functions, from design to development, related to the spacecraft/launch vehicle, payload, ground operations and mission management, embracing all the activities that are needed to accomplish the goal of the mission.

Modern approaches to space missions follow design phases of increasing complexity [12], as shown in Table 1.1. Such phases are some of the most common ones used in space missions design based on NASA and European Cooperation for Space Standardization conventions.

Specifically, from Table 1.1, mission life cycle typically starts with a Phase 0, on which the goals and preliminary mission analysis are assessed. In Phase A, the main mission operations blocks (namely spacecraft and launch operations, payload operations, ground operations and mission management) are formulated, describing the overall scenario for satellite control. Before launch, Phases B, C and D deal with the detailed design and the actual construction of the spacecraft. These include the detailed definition of spacecraft subsystems and payload and their design, development, assembly, integration, and testing,

Table 1.1: Space mission life cycle.

Phase	Definition	Activities
0	Mission analysis and identification	Definition of scientific goals and assessment of potential orbits
A	Feasibility study	Design, development, and testing of mission operations (spacecraft/launch, payload, ground, mission management)
B	Preliminary definition	Definition of systems requirements, interfaces, instruments, mission schedule and costs
C	Detailed design	Building of spacecraft models and tests (structural, thermal, acoustic)
D	Production	Building of the final spacecraft and tests (electrical systems and navigation)
	Launch	
E	Utilization	Mission operations execution
F	Disposal	Termination of the mission

while still iterating the definition of the operations and ensuring all the commands work appropriately. After launch, mission operations as preliminary defined in Phase A are executed in Phase E. Last Phase F deals with the termination and disposal activities of the whole mission.

Designing space missions that aim to visit multiple orbital way-points presents the complication that the targets to be visited, namely their order and/or positions in space, are not known a priori, but are part of the mission definition itself. This increases the complexity of the first phases of the mission operations life cycle (i.e., Phase 0 and A from Table 1.1); specifically, the mission analysis part and feasibility study of the whole mission from the trajectory design point of view.

1.2 Space Mission Analysis

Before going to the details of the design of trajectories that visit multiple orbital targets, this section defines the concept of mission analysis. Space mission analysis usually refers to the mathematical assessment and definition of spacecraft *potential orbits and/or trajectories* that allow the scientific objectives of the mission to be *best fulfilled* [13].

Two aspects are important when addressing space mission analysis:

1. *Multiple* trajectory options are desirable at the preliminary stage of mission life cycle to allow a correct trade-off analysis.

2. The *best* trajectory options according to specific mission-related criteria need to be identified.

Therefore, the scope of mission analysis is to provide to successive design phases of mission life cycle a set of orbital and/or trajectory options that allow to realize the scientific objectives, and, among those, possibly the best options with respect to some criteria. In order to accomplish such scope, almost any space mission analysis problem can be decomposed as a mixture of four different ingredients [14] that are reported in Table 1.2. These are: (1) mathematical models, i.e., the sets of differential equations that affect the spacecraft trajectory; (2) objectives and/or constraints, i.e., quantities that determine the goodness of a mission; (3) design variables, i.e., parameters needed to model the spacecraft trajectory; (4) optimization strategies, i.e., those algorithms and approaches that allow to obtain the desired transfer.

From Table 1.2, the first step for tackling the problem is the definition of a mathematical model that defines the set of differential equations governing the motion of the spacecraft. Among the existing different options (see also Chapter 2), the selection of the mathematical model depends upon the specific mission profile, and it is always a compromise between solution quality and computational effort (see also Chapters 4, 5 and 6). In this sense, the mission analysis problem usually falls within the logic of a multi-fidelity

Table 1.2: Main ingredients that form a mission analysis problem.

Ingredient	Description	Examples
Mathematical models	Set of differential equations that describe the motion of the spacecraft and/or the events that impact such motion	Two-body problem [15, 16], circular restricted three-body problem [16], impulsive thrust model [15, 17], continuous thrust model [18]
Objectives and/or Constraints	Key quantities that determine the goodness of a space mission	Constraint satisfaction [19, 20], optimization [21, 22]
Design variables	Variables sets that describe trajectories relevant for the specific mission objectives/constraints	Direct or indirect methods [23, 24], shooting methods [25, 26]
Optimization strategy	Strategies and algorithms to obtain values for the variables selected in the transfer step to fulfil mission objectives/constraints	Non-linear programming [27], meta-heuristics [28, 29]

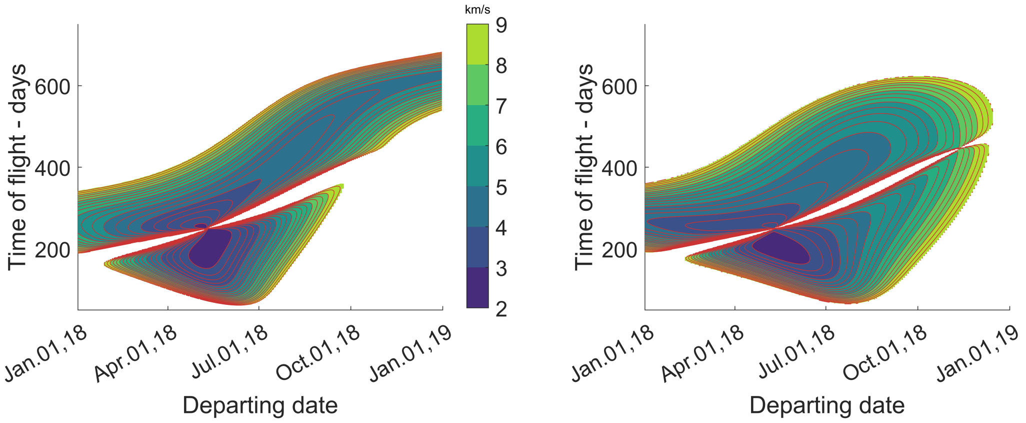
process [30–32], on which mathematical models of different fidelities are employed in different stages of the mission life cycle. As an example of the application of such ingredients, one can consider the design of a mission from Earth to Mars as a typical interplanetary mission analysis problem. In is case, a common practice is to employ the so-called patched conics approximation [15, 16, 25, 33] to model the spacecraft motion (see also Chapter 3). In this example, no other manoeuvres are considered. This approach is proven to be accurate enough for preliminary design and useful for successive refinement in higher-fidelity dynamical models [16].

To fulfill mission analysis' aim , i.e., to provide sets of trajectory options which are compliant with given constraints and optimum with respect to mission-specific criteria, the second ingredient of mission analysis as from Table 1.2 consists in the definition of objectives and/or constraints. These represent key quantities that define the goodness of the mission itself. The mission analysis problem can thus be formulated from two different, though related, formulations that are briefly introduced here and expanded in later Chapters:

- The first one, and more popular within the space trajectory design sector [23, 24], is the spacecraft trajectory optimization formulation [14, 34–36], whereby some mission-specific objective functions are defined and minimized/maximized with respect to some constraints.

- The second one is similar to the so-called constrained satisfaction problem (CSP) formulation [19, 20], on which the aim is to find all the trajectory options that are compliant with given mission-driven constraints.

In the Earth-Mars example, typical objectives to be minimized are the overall mass of the spacecraft, to reduce costs at launch, or the overall mission duration, to reduce the cost of operations. Constraints that are usually considered include departing date and transfer time ranges, launcher performances, and spacecraft propulsion system capabilities. It is very useful to solve an optimization problem in the context of a space mission analysis to understand the minimum performances that one could expect from the spacecraft to fulfill the objectives. Alongside the optimal performances with respect to the aforementioned objectives, on an Earth-Mars transfer one might be interested in gaining more understanding about the flexibility of the mission itself. For example, one might want to identify how many transfer opportunities exist with respect to the given constraints, what are the available launch dates, what are the objectives associated to each one of them, how many trajectory options exist provided different launchers' performances, if the distance from the Sun along any of these options is compatible with power and/or thermal requirements, and so on. This is when the CSP becomes relevant, i.e., when not only optimal trajectories are needed to identify minimum functional performance of the spacecraft, but also when several mission parameters depend upon the trajectory itself, thus different options are needed at a very preliminary stage. In this sense, one should highlight that in this sense the CSP is a feasibility problem that does not represent an alternative to trajectory optimization, but comes along with it to truthfully inform the preliminary mission design with the extent of mission availability and performances.



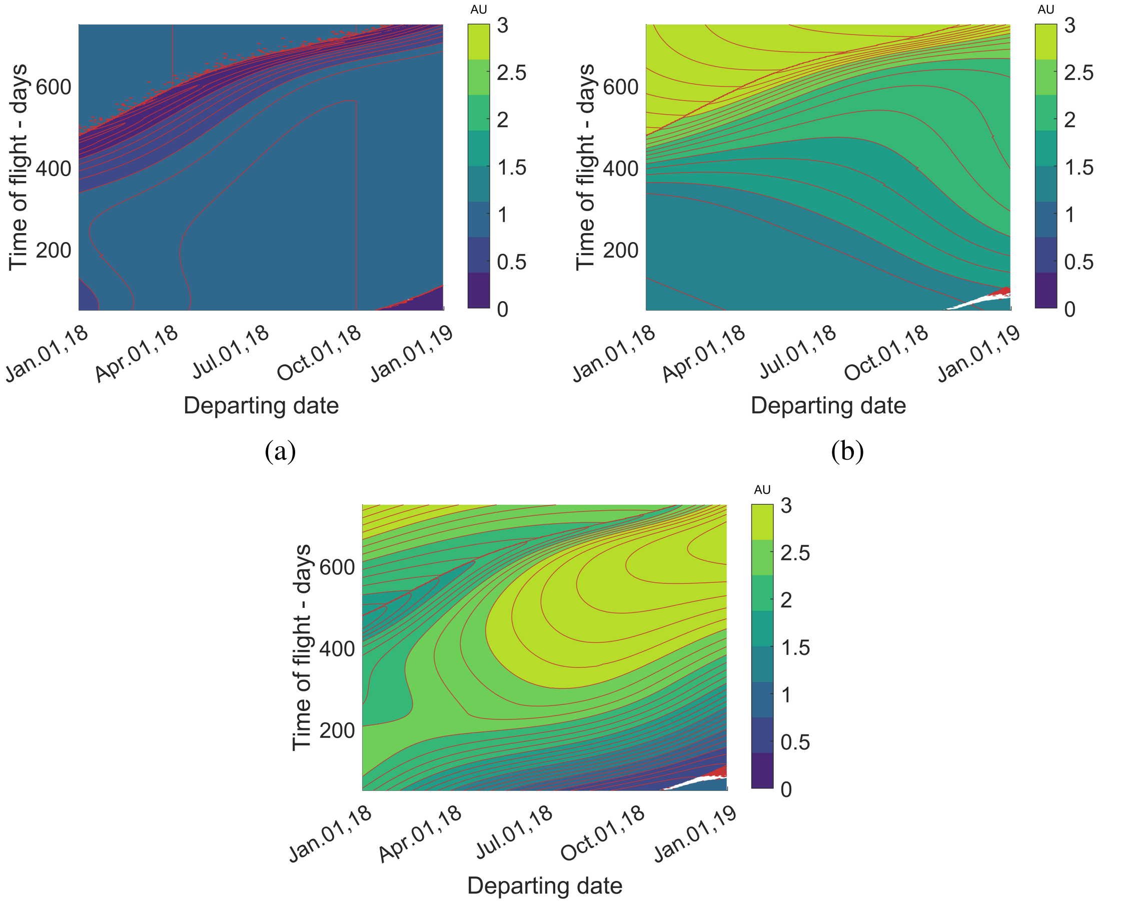
(a)

(b)

Figure 1.2: Impulses maps at Earth departure (a) and Mars arrival (b) for trajectories departing in 2018 window.

The third ingredient from Table 1.2 is about the definition of the orbits themselves that are relevant for the specific mission objectives and constraints identified as second ingredient. It should be noted that this third ingredient does not solve the transfer but rather sets the rules to solve it. In the case of the analysis of the Earth-Mars mission, the orbits relevant for the design (in the planet-to-planet interplanetary transfer) are those that link the two planets (considered as point masses in the patched conics approximation) in a given time of flight. This corresponds to the well-known orbital two-point boundary value problem (TPBVP), commonly known as Lambert's problem, whose solution in two-body dynamics follows well-established procedures [37–40] (see also Appendix A). In successive iterations of the mission analysis (from Phase A), more detailed analysis on the transfers is needed, thus numerical integration within multi-body gravity fields are performed [25].

The last ingredient identified in Table 1.2 to tackle a mission analysis problem consists in the optimization strategies to obtain the solutions (i.e., trajectory options) regarding the chosen design method (first and third ingredients) with respect to the objectives and/or



(a) (b)

(c)

Figure 1.3: Map of minimum (a) and maximum (b) Sun-spacecraft distance, as well as maximum Earth-spacecraft distance (c) for Earth-Mars trajectories departing in 2018 window.

constraints (second ingredient). Since the CSP is relevant for the Earth-Mars example, one can decide to enumerate all the possible departure dates and transfer times (taken within given boundaries) between the Earth and Mars, making them vary discretely on grids, and then solve Lambert's problems for each option identified. In this way, a grid optimization [41] is set up to solve the problem in that it finds multiple trajectory options (all the possible ones in this case) that satisfy mission constraints (on the launch date and transfer times in this example).

A typical output for such a mission analysis problem is shown in Figure 1.2. Here, the maps of impulses at Earth (Figure 1.2.a) and Mars (Figure 1.2.b) are plotted with respect to the departing dates and transfer times, assuming a launch happening in 2018.

Such maps can be used to infer launcher and spacecraft capabilities to arrive at (an orbit around) Mars [15, 16]. Any point in the map represents a trajectory between the Earth and Mars that departs at a given epoch and arrives after given transfer days. The third dimension is completed with the impulses at the two planets (these are computed as velocity changes to match spacecraft velocities with the planets' ones). Since the overall CSP has been solved, one can extrapolate any trajectory-related information that is relevant for other mission design phases. As an example, Figure 1.3 represents the minimum (Figure 1.3.a) and maximum (Figure 1.3.b) spacecraft distance from the Sun for any trajectory option identified, which could be of interest, for example, for designing thermal and power subsystems, alongside the maximum Earth-spacecraft distance (Figure 1.3.c) that is useful for ground communication purposes.

1.3 Mission Analysis of Multi-Target Missions

The Earth-Mars preliminary mission analysis performed in section 1.2 is a typical example of single-target mission. Let's consider now the case of missions that aim to visit multiple targets, which is the scope of the present thesis. Missions that can visit multiple targets (e.g., planets, asteroids, space debris...) by means of one single spacecraft represent the next cornerstone for space exploration and exploitation, as they generally allow to reduce the average cost for visiting each target object.

Designing transfers that visit multiple targets is very similar to the well-known Traveling Salesman Problem (TSP) [42], on which a salesperson must visit several cities, with the aim of maximizing the overall length of the tour. This is one of the most studied problem in combinatorial optimization, and motivated the development of a large class of solutions methodologies in the field of global optimization, from deterministic branch and bound strategies [43,44] to stochastic meta-heuristics [28]. In the classical version of the TSP, the cities to be visited are considered as fixed points in space, and the salesperson must visit each one of them once and return to the origin city, in an overall closed path. Translating the TSP analogy to multi-target missions [45], the cities are typically represented by the objects to be visited. Two main differences can thus be spotted between the classic TSP and the space mission analysis variant:

- The ‘cities’ are not fixed in space, as they are objects subject to forces in space (mainly gravitational ones), thus their position is strictly related to the time at which the spacecraft visits them. In this sense, a typical multi-target mission analysis problem is mixed-integer in nature [46, 47], as it is represented by both integers’ variables, encoding the sequence of targets to be visited, and continuous-varying parameters, i.e., that can vary in domains of real numbers and that usually model mission-specific events such as the launch, transfer times between targets or thrust arcs/manoeuvres.
- The path of visiting the objects usually does not require to return to the initial point, thus the solution space is larger than the classic TSP.

Despite these differences, the analogy with the classic TSP is still informing, as one could make use of the intense literature on the problem to tackle (a part of) the design of multi-target missions (see also later section 1.3.1). In this sense, multi-target mission strategies imply a considerably increase in the complexity of the space mission analysis part within the mission analysis life cycle, as the selection of targets to be visited is part of the trajectory planning itself. In other words, the sequence of targets to be visited is not known a priori (as in the example of Earth-Mars example mission design from section 1.2), but it becomes part of the design.

Such kind of problems has been generally formulated in the form of mixed-integer nonlinear programming (MINLP) [46, 47] also known in literature as hybrid optimal control problem (HOCP) [48]. This is one of the most challenging optimization problems, as it requires the solution of a combinatorial problem mixed with optimal control theory. MINLP/HOCP can be seen as two nested optimization problems: (1) the combinatorial problem, aiming at finding the optimal sequence of targets to be visited, and (2) continuous problem, aiming at identifying one or more locally optimal trajectories for a candidate target sequence in terms of launch date, targets phasing and thrust arcs. The complexity arises from the fact that these two components are generally highly coupled together, or, in other words, while the goodness of a candidate sequence highly depends upon the solution of the continuous optimization, a variation of even a single target will correspond to a significantly different trajectory path, which requires a different set of continuous variables to be optimized.

1.3.1 Exemplar Application: Global Trajectory Optimization Competitions

Given the inherent complexity of multi-target mission design problems, they have been a recurrent proposition in all the editions of the Global Trajectory Optimization Competition (GTOC). GTOCs are international competitions firstly proposed by the Advanced Concept Team (ACT) at ESA in 2005 [49] with the aim of challenging the international scientific community with 'nearly impossible problems' in the field of space trajectory optimization. GTOC challenges usually attract diverse and outstanding teams that propose innovative problems and solutions in the context of complex multi-target missions, making them an opportunity for research in the field of multi-target spacecraft trajectory design and optimization techniques. The complexity of GTOC problems generally adheres to the two following principles [50], in an effort to simulate the uncertainties in preliminary mission design:

- The design space is large on the mixed-integer domain, such that multiple local optima solutions exist. This implies large datasets of targets to be visited (integer variables) with usually many thousands of objects, and, for example, large launch windows, presence of thrust arcs or impulsive manoeuvres, or a combination of the two (continuous variables).

- Objective functions and/or constraints are unusual, so that no canned methods or existing software can solve the problem as a black box.

A breakdown of all the GTOC competitions to date, with problems, winning teams and methodologies are reported in Table 1.3 [2].

From Table 3, one notices that typical solution methodologies for complex multi-target problems usually involve three steps [60,61]:

- Definition of subset of potential targets to be visited (usually done by a pruning process), based on their orbits and scientific characteristics [52,62].
- Employment of global optimization methods (such as enumerative, branch and bound [52,57,58,63], genetic algorithms [64,65], ant colony optimization [57,58]) to scan the whole domain (usually to select promising sequences of objects).
- Employment of either direct [63] or indirect methods [51] and/or global optimization methods (e.g., particle swarm [62]) for further refinement and improvement of a sub-set of solutions (usually once the visiting sequence of targets is fixed).

The second step on an overall multi-target mission design, i.e., the global scan of the search space in terms of targets' sequences, is typically the most critical, as there are always many options available that can potentially be considered. It is often impractical to perform detailed analysis in terms of transfer modelling and solution achievement (see third and fourth ingredients from Table 1.2) for each of the sequence options.

Let's consider for example the case of GTOC9 problem. This particular edition of GTOC challenges aimed to simulate the design of multi-target missions to restore Earth's orbital environment, in answer to the rapid increase of low-Earth orbit (LEO) debris and the related collisions event triggered by the so-called Kessler's syndrome [66,67].

The purpose of the challenge [68] was to design N missions (N to be found by the optimization process) to cumulatively remove up to $N_{set} = 123$ debris objects. Along each

[2] For problems' description, the interested reader is referred to `https://sophia.estec.esa.int/gtoc_portal/`, last accessed September 2022

Table 1.3: Summary of all GTOC competitions to date, with problem description, winners and methodologies employed. No methodology description has been found for winning solution of GTOC3, thus a '$--$' has been included in the table.

GTOC	Year	Problem Description	Winner	Solution and Transfer
1	2005	Gravity assist trajectory for asteroid deflection	Jet Propulsion Laboratory	Enumeration with pruning + gradient-based optimization [49]
2	2006	Multiple rendezvous with asteroids	Politecnico di Torino	Manual selection + indirect optimization [51]
3	2007	Asteroid sample return with Earth gravity assist	Centre National d'Etudes Spatiales	$--$
4	2009	Multiple fly-bys with asteroids	Moscow State University	Branch and bound [52]
5	2010	Multiple fly-bys and rendezvous with asteroids	Jet Propulsion Laboratory	Beam search + direct method [53]
6	2012	Global mapping of Galilean moons	Politecnico di Torino	Enumeration + indirect method [54]
7	2014	Multi-spacecraft exploration of asteroid belt with rendezvous	Jet Propulsion Laboratory	Enumeration, particle swarm optimization, genetic algorithm, ant colony optimization + local optimizer [55]
8	2015	Multi-spacecraft mapping of radio sources in the universe	ESA's ACT and JAXA's ISAS	Beam search + non-linear programming [56]
9	2017	Multi-mission removal of sun-synchronous debris	Jet Propulsion Laboratory	Branch and bound with ant colony optimization + non-linear programming [57]
10	2019	Multi-spacecraft exploration of the Milky Way	NUDT-XSCC	Branch and bound with ant colony optimization + non-linear programming [58]
11	2021	Multi-spacecraft exploration of asteroid belt and generation of Dyson ring	Tsinghua University and Shanghai Institute of Satellite Engineering	Beam search and genetic algorithm + indirect method [59]

of the N missions, a spacecraft is assumed to be released by the launcher vehicle in rendezvous conditions (i.e., matching position and velocity) with one of the pieces of debris of the dataset. It then needs to perform successive rendezvous with as many debris as possible to deliver de-orbit kits to allow their re-entry. The motion of the spacecraft and debris cloud is subject to the two-body gravitational field generated by the Earth, perturbed by the oblateness effect. The objective is to minimize the cost to launch all the N missions (the lower the spacecraft mass, the cheaper the mission). For a detailed description of the challenge, the interested reader is referred to [68].

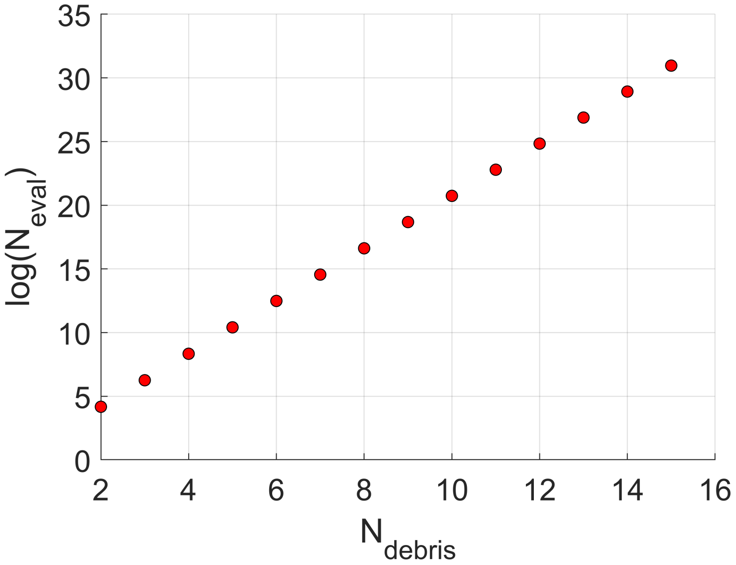


Figure 1.4: Different permutations of debris sequences with respect to the number of debris per sequence. The y axis is in logarithmic scale.

Finding trajectories that link any sequence of debris is a difficult task on its own, involving the solution of complex optimal control problems that are usually time consuming. Thus, considering all the possible sequences of N_{debris} targets imply an amount of:

$$N_{eval} = \frac{N_{set}!}{(N_{set} - N_{debris})!} \tag{1.1}$$

where N_{eval} is the number of multi-target trajectories to be analysed. Figure 1.4 shows the variation of N_{eval} with respect to N_{debris}. For any $N_{debris} = 5$, the number of possible sequences is already approximately 2.59×10^{10}. For reference, in the best solution found by Jet Propulsion Laboratory (JPL) [57], among $N = 10$ missions designed, the shortest sequence found has 9 objects (approximately 4.77×10^{16} permutations), while the longest one has 14 (approximately 8.41×10^{28}). It is thus impractical to optimize in detail each trajectory in preliminary design, and the key to solving this class of problems must lie in managing this complexity efficiently.

A common way to handle such complexity is represented by multi-fidelity approaches [30–32], on which approximate transfer analyses estimate crucial parameters for any target-to-target transfer, such as the propellant consumption (through the Δv) and/or the time of flight (T). These are typically analytical approaches that allow a quick assessment of the goodness of a specific sequence. In this way, the multi-target mission design problem is uncoupled in two steps:

- Usage of low-fidelity approximate formulas to quickly compute Δv and T between two targets.
- Reconstruction of the actual trajectory in high-fidelity model, using the low-fidelity one as initial guess.

Moving the example of the GTOC9 further, one could, for example, consider approximate Δv computations between two pieces of debris by evaluating the change of semi-major axis, inclination, and right ascension of the ascending node. Once such approximation has been performed, the problem of finding sequences of targets could be tackled by means of any global optimization algorithm that come from the combinatorial optimization field [28,43,44]. This approach has been followed by several top-level teams [69,70], consistently resulting in high performance (i.e., low cost) ADR missions. However, the strategy of the JPL team was still able to return better performances, as no approximation was employed, but rather a massive database of debris-to-debris trajectories already optimized with non-linear programming methods was created by means of very high-performance computers [57]. Then, branch and bound and ant colony optimization were used to construct sequences of debris and to assemble the final campaign of N=10 missions.

This suggests that care should be taken when tackling such problems with a multi-fidelity paradigm, since one might want to use the low-fidelity step only when the solutions lie close to the actual ground truth. Experience then suggests that approximate methods [71]:

- Lead to sub-optimal solutions that may lie far to the ground truth (e.g., JPL solutions performs worse when computed in low-fidelity models [71]).
- Using approximate methods to estimate Δv and T that correspond closely to the actual transfers within few percentage of error is a current open research field in the context of multi-target missions [72, 73].

In conclusion, the usage of multi-fidelity framework is proven particularly useful as it disregards the use of massive computer memory (that it is implied by JPL-like brute force approaches) and it provides efficient approximations for a quick scan of the feasible search space. However, one should carefully consider the amount of knowledge to be transferred from one layer of fidelity to the next one. This work, and the related toolboxes (see section 1.5.1), aims at providing insight of the information that is relevant to be passed between two different layers of fidelity, without both losing valuable or imposing artificial/inaccurate knowledge (see Chapter 4 and 5).

1.3.2 Multi-Target Missions and Constrained Satisfaction Problem

Similarly, to the Earth-Mars mission analysis problem in section 1.2, a proper mission feasibility study in the context of multi-target mission design is not only interested in tackling the global optimization problem (on a GTOC-like manner), but also in identifying multiple trajectory options, i.e., in solving the CSP. This is particularly relevant for multi-target missions, and it is again because of the uncertainties of the preliminary design phases. However, the constraint satisfaction problem, alongside the optimization one as discussed in section 1.2, is amplified due to the mixed-integer nature of such missions. In fact, one should consider:

- Given the sequence of targets, to perform a sensitivity analysis on the continuous-varying parameters (similarly to the Earth-Mars case of section 1.2).
- Not knowing the sequence of targets, to retain as many sequence options as possible (i.e., acting on the integer domain) to allow successive design phases a correct trade off.

The relevance of having multiple sequence options could be explained with the following example. Consider the analysis for CASTAway mission [9, 10], on which the author collaborated with a large consortium of scientists around Europe, by supporting the mission analysis part (see section 1.5.5). The aim of CASTAway is to pass-by asteroids with the aim of maximizing the scientific return of a thorough exploration of the MAB. One should notice that such an objective is rather difficult to define, and no clear measurable objective function can be formulated (see also Chapter 2 for more details on typical formulation of objective functions). This is because it is generally complex to set up a cost function that considers crucial (measurable) parameters (like the Δv or the mission duration) with scientific relevance of the targets, alongside observability conditions (e.g., the camera field of view). Therefore, the approach was to identify the largest set possible of different asteroids sequences that fulfill some constraints and that minimize some objectives (i.e., the overall Δv consumption). This is precisely a constraint satisfaction problem.

To conclude, the study of multi-target missions remains partially unexplored and thorough analysis is still required. Such missions would not only have a positive economic impact but would also add remarkable value from a more general space-exploration and scientific perspectives.

1.4 Research Purpose

1.4.1 Aim

As identified in the present Chapter, as well as in Chapter 2, there is a need for the initial design of multi-target missions from the two different, though related, formulations of a typical mission analysis problem, i.e., the global optimization problem and the constraint satisfaction problem, in an answer to the uncertainties arising during the preliminary mission design phases.

To do so, this thesis tackles typical interplanetary trajectory design problems, namely trajectories with multiple gravity assists (Chapter 5) and visiting multiple asteroids (Chapter 6). However, it should be noted that approaches developed in this work are extendable to any other multi-target mission (not necessarily related to interplanetary travels).

As such, the aim of the present thesis can be summarized as follows:

To establish an efficient and robust pipeline to tackle both the global optimization problem and the constraint satisfaction problem for space missions that aim to visit multiple targets.

1.4.2 Objectives

To fulfil the declared aim, the following objectives are identified:

i. To review the current trends for space mission analysis in the context of multi-target trajectory design, and relate such context to optimization and constraint satisfaction fields.

ii. To assess the feasibility of different model types on an overall multi-fidelity design of multi-target trajectories.

iii. To propose efficient strategies for the global exploration of the search space for multi-target missions.

iv. To tackle the constraint satisfaction problem for multi-target missions.

v. To demonstrate the efficiency and robustness of such approaches with relevant and practical test cases.

Section 1.5.1 contextualizes the defined objectives within the present thesis and relates to the proposed novelties and outcomes.

1.5 Research Contribution and Thesis Structure

The research contained in the present work has been distributed in the form of different conference papers, presentations, and journal publications. Moreover, relevant toolboxes (encoded in MATLAB) have been developed. One should notice that all the work that is presented in this thesis was carried out and developed entirely by the main author, while the supervisors contributed to the work with their priceless guidance and advice.

The following sub-sections describe the thesis structure, alongside the main outcomes and novelties produced with the present research (section 1.5.1), alongside the main toolboxes developed to carry on the research (section 1.5.2), as well as the relevant conference papers, presentations, journal articles (sections 1.5.3 and 1.5.4) and technical reports for several mission proposals in collaboration with large consortium of scientists (section 1.5.5). The author also helped in the supervision of several theses in the context of the Space and Astronautical Engineering M.Sc. at Cranfield University (section 1.5.6).

1.5.1 Outcomes, Novelties and Structure of the Thesis

The literature review discussed in Chapter 2 shows clear research interests in the definition of a proper framework for tackling multi-target mission designs, that is robust and efficient within the context of interplanetary trajectories. In fact, the present thesis mainly deals with the mission analysis of multi-target trajectories that aim to visit either multiple planets, on overall multiple-gravity assist (MGA) transfers, or multiple asteroids, or a combination thereof.

One has thus identified the main original contributions to literature:

- The global optimization problem both from the single-objectives and the multi-objective point of view have been tackled with an efficient and robust procedure based on exploitation of graph theory and dynamic programming principles (Chapter 5).
- A CSP-like approach for missions that visit multiple asteroids is tackled alongside the global optimization problem by extending the dynamic programming principles exploited for previous points. The paradigm of dynamic programming as described in Appendix B is modified allowing the selection of multiple paths at each decision step to provide large set of sequences that fulfil mission-specific constraints (Chapter 6). In this way, more and diverse solutions are retained in the final solutions set compared to standard approaches (like BS and standard dynamic programming).

Table 1.4 describes the thesis structure and its relationship with the objectives fulfilled, and the identified outcomes and novelties.

Table 1.4: Breakdown of the present thesis with respect to chapters, objectives fulfilled and outcomes/novelties.

Chapter	Objectives Fulfilled	Description	Outcomes and Novelties
2: Trajectory Design Approaches for Multi-Target Missions	i	Understand the existing literature and identify the gaps. The pipeline for tackling multi-target interplanetary missions is presented.	The literature gaps are identified. A taxonomy on mission analysis problems is also presented.
3: Modelling Interplanetary MGA Trajectories	ii	Set the mathematical model used in the thesis.	Extend the theory on existing mathematical models for MGA trajectories and apply to complex test cases.
4: MGA Sequences via Tisserand Graph	ii, iii	Describe how to exploit a Tisserand graph for designing MGA trajectories.	The relationship between different mathematical models is assessed (from circular-coplanar to inclined-eccentric planetary orbits). Tisserand graphs are exploited for constructing high-inclination orbits.
5: Multi-Objective MGA Design via Dynamic Programming	ii, iii, v	Graph theory is exploited to efficiently explore the search space in the context of MGA trajectory design.	Concepts of dynamic programming are extended and applied to the single-objective and multi-objective optimization of MGA trajectories. The relationship between different mathematical models is assessed. The design of highly inclined trajectories is also discussed.
6: Modified Dynamic Programming for MAB Exploration	ii, iii, iv, v	The problem of visiting multiple asteroids while using MGA with planets is tackled.	Concepts of dynamic programming are extended and applied to the constraint satisfaction problem for asteroids exploration.
7: Conclusions	−−	Summary of the work is presented, alongside final remarks, current limitations, and future work.	−−

1.5.2 Toolboxes and Applications

Alongside the novel contribution to the existing literature, an important outcome of the present research is the production of codes and toolboxes that could be ideal for the design of interplanetary multi-target missions (both MGA and exploration of asteroid tours). These are deliverable for one of the sponsors of the present research project, namely Airbus Defence and Space [3].

The functioning of such toolboxes alongside practical examples are discussed in detail in Appendix C. The toolboxes can be found at [74]. This section provides an overview of the main blocks that are developed within the rest of the present work. Table 1.5 presents the main blocks of the developed codes, also contextualizing them within the present thesis, while Figure 1.5 shows the relations between such blocks, also providing a practical description.

Table 1.5: Breakdown of the present thesis with respect to chapters, objectives fulfilled and outcomes/novelties.

Toolbox	Definition	Chapter
AUTOMATE	AUTOmatic Multiple-gravity Assist with Tisserand Exploration	4
ASTRA	Automatic Swing-by TRAjectories	5
RESTOUR	RESonances TOUR	4, 5
DYNAMIS	DYNamic programming for Asteroid MISsions	6

As it can be seen from Figure 1.5, the toolboxes follow the same logic of multi-fidelity paradigm that is usually employed for designing multi-target missions. For example, one could consider CASTAway [9, 10], Dolphin [75] or Icarus [76] mission design problems. The author provided actual support (see also section 1.5.5) for Phase 0 proposals' preparation of such missions in the context of ESA's M7 and F2 [77] mission call. In particular: CASTAway focuses on the exploration of the MAB region with the aim of passing-by 10 asteroids; Dolphin aims at maximising the heliocentric inclination to analyse the interplanetary and interstellar dust particles; Icarus goal is to rendezvous with a low-perihelion

[3] This work was supported by Airbus Defence and Space through the Cranfield University Industrial Partnership Framework (MITnTargets: Mixed-Integer Trajectory Design for Large Number of Targets)

asteroids to observe the disruption mechanisms when passing-by the Sun. Each tool from Table 1.5 has been used to support the mission analysis and their relevance is highlighted with the following discussion.

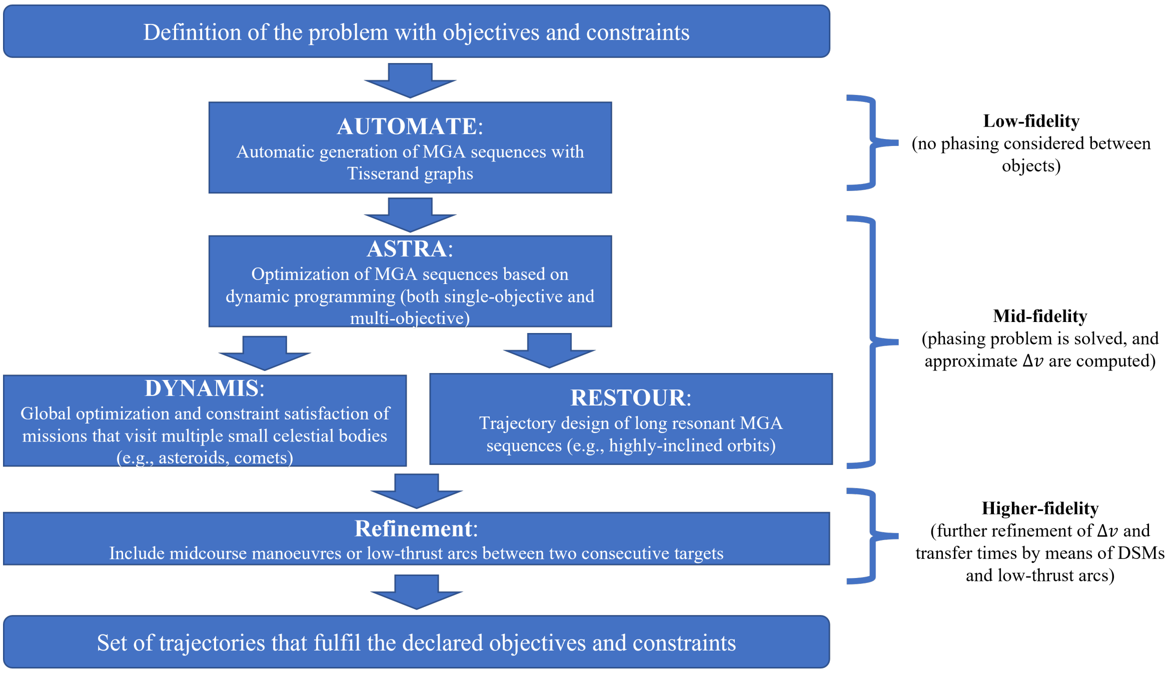

Definition of the problem with objectives and constraints
AUTOMATE: Automatic generation of MGA sequences with Tisserand graphs
ASTRA: Optimization of MGA sequences based on dynamic programming (both single-objective and multi-objective)
DYNAMIS: Global optimization and constraint satisfaction of missions that visit multiple small celestial bodies (e.g., asteroids, comets)
RESTOUR: Trajectory design of long resonant MGA sequences (e.g., highly-inclined orbits)
Refinement: Include midcourse manoeuvres or low-thrust arcs between two consecutive targets
Set of trajectories that fulfil the declared objectives and constraints
Low-fidelity (no phasing considered between objects)
Mid-fidelity (phasing problem is solved, and approximate Δv are computed)
Higher-fidelity (further refinement of Δv and transfer times by means of DSMs and low-thrust arcs)

Figure 1.5: Summary of developed toolboxes with respect to input/output relations and multi-fidelity framework.

The preliminary mission analysis usually starts with the definition of the space mission in terms of objectives and constraints to be fulfilled. At a first level of fidelity, AUTOMATE is used to quickly assess the feasibility of different planetary sequences of reaching the desired orbital region, without explicitly solving the phasing problem, but rather providing energetic viability. In the case of CASTAway, Dolphin, and Icarus these are the MAB region, high-inclination orbits, and low-perihelion orbits, respectively.

The core of the software is ASTRA, that takes the planetary sequences (and the encounter conditions with the planets) from AUTOMATE and tries to find suitable trajectories in terms of planetary phasing and approximate Δv by exploiting dynamic programming principles (both single-objective and multi-objective optimization could be performed).

Using the trajectories found by ASTRA, DYNAMIS is then used to check the accessibility of asteroid encounters, by solving both the global optimization and the constraint satisfaction problems (if requested). This is particularly useful for CASTAway-like missions, on which multiple asteroid fly-bys are to be designed.

Also taking input from ASTRA, RESTOUR is useful to solve the combinatorial problem associated with MGA resonant tours (Chapter 4 and 5). This toolbox was developed and used by the author when working on the mission analysis for Dolphin proposal, aiming at maximizing the heliocentric inclination to observe interstellar and interplanetary dust particles. Finally, trajectories coming from ASTRA, DYNAMIS and RESTOUR can be refined in more detailed models taking into account, for example, of the presence of mid-course manoeuvres, i.e., the so-called Deep Space Manoeuvres (DSMs), or low-thrust arcs.

The final output is a set of trajectories that fulfill the declared objectives and constraints, and related crucial information, like the launch window, propellant consumption, transfer times, magnitude of thrust and so on.

1.5.3 Conference Papers and Presentations

I. AFSA, H., **Bellome, A.**, Sánchez, J.P., Kemble, S., ‘Automatic Multi-Gravity Assist Trajectory Design with Modified Tisserand Graphs Exploration’, conference paper and interactive presentation at 73rd International Astronautical Congress, Paris, France, (September 2022)

II. Sánchez, J.P., **Bellome, A.**, Del Ser, J., Carrillo, M., ‘Deterministic and stochastic exploration of long asteroid fly-by sequences exploiting tree-graph and optimal sub-structure properties’, conference paper and oral presentation at 73rd International Astronautical Congress, Paris, France, (September 2022)

III. **Bellome, A.**, Sánchez, J.P., Rico Alvarez, J.I., AFSA, H., Kemble, S., Felicetti, L., ‘An Automatic Process for Sample Return Missions Based on Dynamic Programming Optimization’, conference paper and oral presentation at AIAA SciTech 2022 Forum, San Diego, California, United States of America, (January 2022)

IV. **Bellome, A.**, Carrillo, M., Sánchez, J.P., Del Ser, J., Kemble, S., Felicetti, L., ‘Efficiency of Tree-Search like Heuristics to Solve Complex Mixed-Integer Programming Problems Applied to the Design of Optimal Space Trajectories’, conference paper and oral presentation at 72nd International Astronautical Congress, Dubai, United Arab Emirates, (October 2021)

V. **Bellome, A.**, Sánchez, J.P., Kemble, S., Felicetti, L., ‘A Multi-Fidelity Optimization Process for Complex Multiple Gravity Assist Trajectory Design’, conference paper and oral presentation at 8th International Conference on Astrodynamics Tools and Techniques, Virtual Event, (June 2021)

VI. **Bellome, A.**, Nakhaee-Zadeh, A., Zaragoza Prous, G., Leng, L., Coyle, M., D’Souza, S., Mummigatti, S., Serfontein, Z., ‘Applications of Nanosatellites for Lunar Missions’, conference paper and oral presentation at IEEE Aerospace Conference (AeroConf2021), Big Sky, Montana, United States of America, (March 2021)

VII. **Bellome, A.**, Sánchez, J.P., Felicetti, L., Kemble, S., ‘Modified Tisserand Map Exploration for Preliminary Multiple Gravity-Assist Trajectory Design’, conference paper and oral presentation at 71st International Astronautical Congress, The Cyberspace Edition, Virtual Event, (October 2020)

1.5.4 Journal Publications

I. **Bellome, A.**, Sánchez, J.P., Felicetti, L., Kemble, S., ’Design of Asteroids Tours with Modified Dynamic Programming’, *Acta Astronautica* (manuscript in preparation)

II. **Bellome, A.**, Sánchez, J.P., Felicetti, L., Kemble, S., ‘Multi-Objective Design of Gravity Assist Trajectories via Graph Transcription and Dynamic Programming’, Journal of Spacecraft and Rockets, 1-19, 2023, `https://doi.org/10.2514/1.A35472`

III. Armellin, R., Beauregard, L., **Bellome, A.**, Bernardini, N., Fossà, A., Fu, X., Parigini, C., Pirovano, L., Wijayatunga, M., 'Team theAntipodes: Solution Methodology for GTOC11', Acta Astronautica 201, 142-151, 2022, `https://doi.org/10.1016/j.actaastro.2022.08.034`

IV. Lehtinen, T., Granvik, M., **Bellome, A.**, Sánchez, J.P., 'Icarus: In-Situ Monitoring of the Surface Degradation on a Near-Sun Asteroid', Acta Astronautica 186, 98-108, 2021, `https://doi.org/10.1016/j.actaastro.2021.05.028`

1.5.5 Technical Reports

I. Howett, C., Nowicki, K., Calcutt, S., Sánchez, J.P., **Bellome, A.**, Bewick, C., Rühl, T., Bowles, N., Antti, N., Kohout, T., Pommerol, A., Thomas, N., Jørgensen, J. L., Preston, L., Snodgrass, C., Rivkin, A., Jones, G., King, A., 'CASTAway - Mapping the Evolution of our Solar System: A Candidate for the ESA M7 Mission'

II. Sterken, V., Agarwal, J., Baalmann, L. R., **Bellome, A.**, Briois, C., Della Corte, V., Dialynas, K., Guidi, G., Horanyi, M., Huybrighs, H., Herbst, K., Hunziker, S., Krüger, H., Lasue, J., Li, A., May, B., Richardson, J., Rowan-Robinson, M., Sánchez, J.P., Schönbächler, M., Srama, R., Trieloff, M., 'DOLPHIN - The Dust Observatory to study the LIC, interPlanetary dust, and Heliospheric Interactions in our Neighborhood: An F-class mission proposal to ESA'

III. Barabash, S., Bolin, B., Britt, D., Futaana, Y., Hagermann, A., Holt, C., Karatekin, Ö., Kero, J., Laufer, R., Mann, I., Michel, P., Molaro, J., Muinonen, K., Norberg, O., Palomba, E., Pokorny, P., Richardson, D., Shimoyama, M., Srama, R., Szalay, J. R., Trieloff, M., Virkki, A., Walsh, K., Wiegert, P., Ye, Q., **Bellome, A.**, Bottke, W. F., Brown, P. G., Granvik, M., Herdrich, G., Jedicke, R., Kastinen, D., Knight, M., MacLennan, E., Masiero, J., Mocker, A., Moskovitz, N., Neumann, W., Näsiläm A., Penttilä, Pommerol, A., Sánchez, J.P., Snodgrass, C., Sterken, V., Toliou, A., Tsirvoulis, G., Vokrouchlicky, D., Wang, X., Wieser, M., 'Icarus - Witnessing the demise of a near-Sun asteroid: Phase-1 proposal in response to ESA's call for the F2 mission'

1.5.6 Master Thesis Supervision

I. Yunta, J., '*Multi-Target Rendezvous Combinatorial Optimisation Applied to GTOC9*', Thesis within the course 'Astronautics and Space Engineering M.Sc.', Cranfield University, 2019-2020. Main supervisor: Dr. Joan-Pau Sánchez

II. Tena, I., '*Global Optimisation for Large Combinatorial Space Trajectory Design Problems: CASTAway Case Study*', Thesis within the course 'Astronautics and Space Engineering M.Sc.', Cranfield University, 2019-2020. Main supervisor: Dr. Joan-Pau Sánchez

III. Gallego, I., '*Multi Gravity Assist Trajectories Preliminary Analysis with Tisserand Graphs*', Thesis within the course 'Astronautics and Space Engineering M.Sc.', Cranfield University, 2019-2020. Main supervisor: Dr. Joan-Pau Sánchez

IV. Rico, J. I., '*Search Space Pruning Of Candidates Applied To Comet Sample Return Mission Trajectory Design*', Thesis within the course 'Astronautics and Space Engineering M.Sc.', Cranfield University, 2020-2021. Main supervisor: Dr. Joan-Pau Sánchez

V. AFSA, H. '*AUTOMATE: Automatic Multi-Gravity Assist Trajectory Design with Tisserand Exploration*', Thesis within the course 'Astronautics and Space Engineering M.Sc.', Cranfield University, 2020-2021. Main supervisor: Dr. Joan-Pau Sánchez

VI. Carrillo, M. '*Stochastic Optimisation for Complex Mixed-Integer Programming Problems in Asteroid Tour Missions*', Thesis within the course 'Astronautics and Space Engineering M.Sc.', Cranfield University, 2020-2021. Main supervisor: Dr. Joan-Pau Sánchez

Chapter 2

Trajectory Design Approaches for Multi-Target Missions

This Chapter provides an overview of background theory and methods for mission analysis in the context of multi-target missions. Section 2.1 of this Chapter first presents the main building blocks that are needed for any mission analysis problem. Two main research areas are then analyzed that are relevant for the next Chapters: multiple-gravity assist (MGA) trajectory design, which is discussed in section 2.2, and multiple asteroids tours, assessed in section 2.3.

The purpose of the present Chapter is thus: (1) to contextualize this thesis within the current literature, (2) to identify existing methodologies and trends for designing multi-target missions, (3) to identify the main research gaps that motivate the present work and (4) to introduce the novelties presented to fill these gaps.

2.1 Ingredients for Mission Analysis

In this section, the main building blocks are described that allow to tackle any mission analysis problem, expanding from Chapter 1 and highlighting those that are significant for this thesis. The ingredients follow the same logic presented in [14, 23, 24, 36] and build on these, especially in the discussion provided in section 2.1.4, that presents typical ways of solving mixed-integer trajectory design problems.

2.1.1 Mathematical Models

The first ingredient to tackle any mission analysis problem consists in defining the mathematical models. These correspond to a set of differential equations that describe the motion of the spacecraft and all the events that affect such motion. The equations of spacecraft motion can generally be written as [21]:

$$\dot{\vec{x}}(t) = f(\vec{x}(t), \vec{u}(t), t) \tag{2.1}$$

where t is the elapsed time since epoch, e.g., the launch date, $\vec{x}$ is is the time history of the spacecraft state vector, $\dot{\vec{x}}$ is its first-time derivative, and $\vec{u}$ is the time history of the control vector. The state $\vec{x}$ typically encodes the position and velocity vectors $[\vec{r}, \vec{v}]$ of the spacecraft alongside its mass $m(t)$. The control $\vec{u}$ usually defines the action of a thrust system, characterized by the exhaust velocity of the propellant mass v_e and by a thrust vector $\vec{T}(t)$.

The choice of a mathematical model depends upon the specific mission application, and usually it is a compromise between computational effort and solution quality. Nevertheless, a mathematical model that is relevant for spacecraft trajectory design usually comes with the definition of the following steps:

- List of celestial objects that affect the spacecraft motion
- Environmental effects such as solar radiation pressure or atmospheric drag
- Representation of the state variables
- Thrust model employed

A very general representation of the system from Eq. 2.1 could be:

$$\begin{cases} \dot{\vec{r}} = \vec{v} \\ \dot{\vec{v}} = \ddot{\vec{r}} = \vec{a}_g + \vec{a}_p + \frac{\vec{T}(t)}{m} \\ \dot{m} = -\frac{\vec{T}(t)}{v_e} \end{cases} \tag{2.2}$$

Eq. 2.2 describes the spacecraft motion under the effect of different actions:

- Gravitational acceleration $\vec{a}_g$ due to the presence of n celestial bodies (e.g., Sun, planets, moons, asteroids) with masses m_i and positions $\vec{r}_i \forall i = 1, ..., n$, in the form of $\vec{a}_g = -G\sum_{i=1}^{n} \frac{m_i(\vec{r}-\vec{r}_i)}{|\vec{r}-\vec{r}_i|^3}$.
- Perturbative accelerations $\vec{a}_p$ due to, for example, irregular gravity fields of the close-by celestial objects, radiation pressures, electro-magnetic actions.
- Acceleration $\frac{\vec{T}(t)}{m}$ due to the propulsive system mounted on board.

Typically, at preliminary stages of mission analysis, it suffices to consider approximate models for the dynamics. A very useful model is the well-known restricted two-body problem, that considers just two dimensionless masses, one being much more massive than the other, that are affected by their mutual gravitational attraction [15]. More detailed models take into consideration orbital perturbations such as the non-sphericity of the gravitational field, presence of other bodies, radiation pressure, or even magnetic fields. These can be relevant to describe specific mission applications, for example missions around the Earth [78], or other irregular bodies, such as moons [79] or asteroids [80], or interplanetary transfers [81].

A very useful extension of the two-body approximation is represented by the circular-restricted three-body problem [16], on which two main attracting bodies, having circular co-planar paths, are assumed to affect the spacecraft motion at the same time (e.g., the Earth and the Moon). This has been extensively used to design missions that take advantage of the presence of the secondary body to achieve the mission objectives, such as [82–86]. Nevertheless, as as anticipated in Chapter 1, a complete study, that goes beyond the preliminary mission analysis, considers relatively soon in the design phases a full-force gravitational model with n bodies.

Within the definition of the mathematical models, one should define the spacecraft state representation. It should be noted that the system presented in Eq. 2.2 assumes the spacecraft state representation $\vec{x}(t)$ as composed by the position and velocity vectors and mass of the spacecraft varying over time. Such vectors (with three components each) are generally referred to an inertial reference frame that depends upon the mission application. For example, for interplanetary missions it is useful to refer the spacecraft state to a frame centered in the Solar System barycentre and axes pointing towards 'fixed stars', or, for Earth-centered missions, the so-called Earth Centered Inertial (ECI) frame could be used [16]. Besides inertial coordinates, many other representations of the state vector can be found in literature [87], such as classical or equinoctial orbit elements, fast or slow variables or canonical variables. The reason for employing one representation with respect to another one might vary from the specific mission application, level of details required for the study, physical meaning, or singularities. For example, some models might benefit from state and control representation that consider classical orbital elements (COEs) that are derived from the position and velocity vectors. COEs are a list of parameters $(a, e, i, \Omega, \omega, \theta)$ that describe the spacecraft orbit through the semi-major axis (a), eccentricity (e), inclination (i), right ascension of the ascending node (Ω), argument of periapsis (ω) and the position of the spacecraft in such orbit through the true anomaly (θ). The main advantage of modelling the state with COEs is that a physical interpretation of the time-variation of dimension, size and orientation of the spacecraft orbit is directly available, and consequently the effect of the thrust (expressed in terms of radial, tangential, and normal components) on the COEs is apparent. The differential equations that describe the time evolution of the COEs are the so-called Gauss Variational Equations (GVEs) [16]. The main limitation of GVEs is that it is affected by singularities at zero inclination, at which the right ascension of the ascending node is not defined, and at zero eccentricity, at which the argument of perigee and the true anomaly are indistinguishable. Therefore, other elements might be useful, such as the so-called modified equinoctial orbital elements, that do not suffer from singularities and are robust to uncertainties, but their physical meaning is not immediately evident. An example of the time variation of the modified equinoctial elements can be found in [34].

Finally, mathematical models should describe the type of thrust models that is employed for the mission analysis (if any). The thrust is generally applied by means of a propulsive system that ejects a propellant from the spacecraft causing an equal and opposite force to the spacecraft, and thus an acceleration. The dynamical system as from Eq. 2.1 can assume different forms if the thrust is applied as:

- Impulsive manoeuvres, or
- Continuous manoeuvres

A manoeuvre can be seen as a modification of the spacecraft state vector by means of an event that occurs along the trajectory (e.g., switching on the propulsion system). In the case of impulsive manoeuvres, one assumes that the spacecraft modifies its velocity instantaneously, through a velocity increment $\Delta\vec{v}$, leaving the position unchanged. This model is usually sufficiently accurate for many practical mission options that employ engines with relatively low v_e and high-thrust, and it has been studied extensively for many different space trajectory design problems, such as [88–91]. The added simplification implied by the impulsive model is that the controls are zero $\vec{u}(t) = 0$. Thus, the spacecraft is only subject to natural external forces except for the time instants at which the manoeuvre is performed to change the spacecraft velocity. On the other hand, continuous manoeuvres are usually suitable for missions that require high v_e and low thrust, thus making the transfer time much larger than the impulsive case. In this case, the control input is not zero $\vec{u}(t) \neq 0$ and therefore the equations of motion as from Eq. 2.2 should consider the thrust term as well.

For the purposes of the present thesis, the mathematical models employed are based on restricted two-body dynamics, inertial coordinates, i.e., position and velocity, for defining the spacecraft state and impulsive thrust approximation.

2.1.2 Objectives and Constraints

The second building block of mission analysis is the definition of mission objectives and constraints. These help identifying mission performances and requirements that define the feasibility and success of the mission itself. Both objectives and constraints are provided in the form of some functions of the state, control, and time $(\vec{x}(t), \vec{u}(t), t)$ and usually relate to (piecewise) continuous-varying quantifiable properties of the mission (like the propellant consumption, transfer time, state errors, and so on).

One should define the following points to be considered in the analysis when addressing the second building block of a general mission analysis problem:

- The type of objective/constraints functions. These are usually mission-related functions that help defining the minimum functional performances that the trajectory must fulfil for the mission to be successful.
- The number of objective functions. The goodness of a mission analysis usually depends upon different criteria, that are typically conflicting one to the other. The problem should be set up in a way that at least the minimum set of performances is considered when designing a mission (typically, this relates to the control effort, or the time required to accomplish the mission [24]).
- The type of problem to be assessed. Generally, mission analysis problems can be formulated as trajectory optimization problems [14, 23, 24, 35, 36], on which the mission designer wants to find one or more trajectories that minimize or maximise specific objective function(s). However, one might want to identify possibly multiple options that satisfy given mission-related constraints, thus addressing the so-called constraint satisfaction problem (CSP) [20].

A general representation of objective functions is given in the Bolza form [92]:

$$J = \Phi(\vec{x}(t_0), \vec{x}(t_f), t_0, t_f) + \int_{t_0}^{t_f} L(\vec{x}(t), \vec{u}(t), t)dt \tag{2.3}$$

where t_0 and t_f are the initial and final time, respectively. Eq. 2.3 highlights the two types of functions that can be considered when analysing space missions. The first function $\Phi(\vec{x}(t_0), \vec{x}(t_f), t_0, t_f)$ is the so-called Mayer term that identifies the objective function either at the initial state or final state, or both, thus it does not depend upon the history of the trajectory itself. The second function $L(\vec{x}(t), \vec{u}(t), t)$ is the so-called Lagrange term that traces objectives as function of the states and controls throughout their time history.

Examples of objective functions that are in the form of the only Mayer term are the total transfer time [93, 94]:

$$J = t_f - t_0 \tag{2.4}$$

the total velocity increment expressed as the sum of the number n_{man} of $\Delta\vec{v}$ needed along the transfer [17, 95–98] [1]:

$$J = \sum_{i=1}^{n_{man}} |\Delta\vec{v}_i| \tag{2.5}$$

or even initial/final conditions [99, 100]. The objective function in Eq. 2.5 is very useful in mission analysis as it is related to the propellant consumption through the well-known rocket equation [16], and thus it has an impact on the mass of the spacecraft.

On the other hand, Lagrange-type functions are integrals of states and controls. Among the most popular functions one has for example the propellant consumption within the so-called fuel-optimal problem in the context of low-thrust trajectory optimization [101–

[1] Note that each $\Delta\vec{v}_i$ is a function of the state variables

103]:

$$J = \int_{t_0}^{t_f} -\frac{|\vec{T}(t)|}{v_e} dt \tag{2.6}$$

Other kinds of objectives can be considered that are specific to the mission application, like spacecraft constellations design [104] (Meyer type), MGA trajectories [95] (Meyer type) or Earth-observation missions [105] (Lagrange type). A typical example of Meyer-type cost function that is used in the present work for MGA trajectories (similarly to Eq. 2.5) is the following:

$$J = v_{\infty,dep} + \sum_{i=1}^{n_{man}} |\Delta\vec{v}_i| + v_{\infty,arr} \tag{2.7}$$

which represents the overall Δv consumption. Eq. 2.7 also accounts for the excess velocities at launch and arrival planets ($v_{\infty,dep}$ and $v_{\infty,arr}$, respectively) and intermediate manoeuvres ($|\Delta\vec{v}_i|$).

Regarding the type of problem to be solved, mission analysis typically addresses it as a trajectory optimization problem, in the form of:

$$\begin{aligned}
&\text{Minimize: } J = \Phi(\vec{x}(t_0),\vec{x}(t_f),t_0,t_f) + \int_{t_0}^{t_f} L(\vec{x}(t),\vec{u}(t),t)dt && \text{(objective function)}\\
&\text{Subject to: } \dot{\vec{x}}(t) = f(\vec{x}(t),\vec{u}(t),t) && \text{(dynamics equation)}\\
&\qquad \vec{x}(t_0) = \vec{x}_0 && \text{(initial conditions)}\\
&\qquad g_i(\vec{x}(t),\vec{u}(t),t) = 0, \forall i = 1,...,m_{eq} && \text{(equality constraints)}\\
&\qquad g_i(\vec{x}(t),\vec{u}(t),t) \geq 0, \forall i = m_{eq}+1,...,m_{in} && \text{(inequality constraints)}
\end{aligned} \tag{2.8}$$

where the functions $g_i(\vec{x}(t),\vec{u}(t),t)$ represent the constraints of the problem at hand (e.g., overall mission duration or Δv), and and are the cardinalities for equality and inequality constraints respectively. Such functions $g_i(\vec{x}(t),\vec{u}(t),t)$ are generally of the following two types [25, 106]:

- Terminal or event-type constraints. These are functions evaluated either at a given event time t_e:

  $$g(\vec{x}(t_e),\vec{u}(t_e),t_e) \geq 0 \tag{2.9}$$

  Or functions evaluated over the trajectory, similarly to the representation in Eq. 2.3:

  $$g_1(\vec{x}(t_f),\vec{u}(t_f),t_f) + \int_{t_0}^{t_f} g_2(\vec{x}(t),\vec{u}(t),t)dt \tag{2.10}$$

- Path constraints, specifying the states and controls over regions of the trajectory. A typical path constraint as from Eq. 2.8 is $\dot{\vec{x}}(t) = f(\vec{x}(t),\vec{u}(t),t)$, that expresses the fact that the spacecraft needs to follow the dynamics equations.

Regarding the number of objective functions, the optimization problem can be either single-objective or multi-objective.

- In single-objective optimization, the goal is to find a set $(\vec{x}(t)^*,\vec{u}(t)^*,t^*)$ that satisfies the constraints, and it is such that:

  $$J(\vec{x}(t)^*,\vec{u}(t)^*,t^*) \leq J(\vec{x}(t),\vec{u}(t),t) \tag{2.11}$$

  for any choice of $(\vec{x}(t),\vec{u}(t),t)$.

- In multi-objective optimization [107], the aim is minimize/maximize a vector function $F(\vec{x}(t),\vec{u}(t),t)) = [J_1,...,J_{n_{obj}}]$ with n_{obj} being the number of objective functions all in the form of Eq. 2.3, usually representing conflicting criteria (e.g., the mission lifetime from Eq. 2.4 and the overall Δv consumption from Eq. 2.5). The solution to the multi-objective optimization problem is the definition of an $(n_{obj}-1)$-dimensional hyper-surface [108] that is the Pareto-optimal set, also known as Pareto front, Pareto frontier or non-dominated front [109]. A solution $(\vec{x}(t)^*,\vec{u}(t)^*,t^*)$

belongs to the Pareto front if there is no other solution that improves all the objectives simultaneously, i.e., it is such that:

$$J_i(\vec{x}(t)^*, \vec{u}(t)^*, t^*) \leq J_i(\vec{x}(t), \vec{u}(t), t) \tag{2.12}$$

for any choice of $(\vec{x}(t), \vec{u}(t), t)$. The multi-objective formulation of a mission analysis problem is very helpful in identifying trade-off between mission critical parameters. An example of Pareto front is shown in Figure 2.1 for a multiple-gravity assist mission from Earth to Jupiter and passing-by Venus, Earth, Mars, and Earth, like the JUICE mission [6]. Two objectives are minimized at the same time, namely the total Δv, representing the manoeuvres required during the mission (also accounting for the spacecraft-planet relative velocities at the Earth launch and Jupiter arrival) and the overall transfer time. A Pareto front like the one in Figure 2.1 is very informative for the mission analysis, showing that trajectory options from Earth to Jupiter exist for relatively short transfer times ($\approx 4-5$ years) but at substantially higher total Δv, with respect to the minimum $\Delta v = 8.948$ km/s requiring 6.450 years.

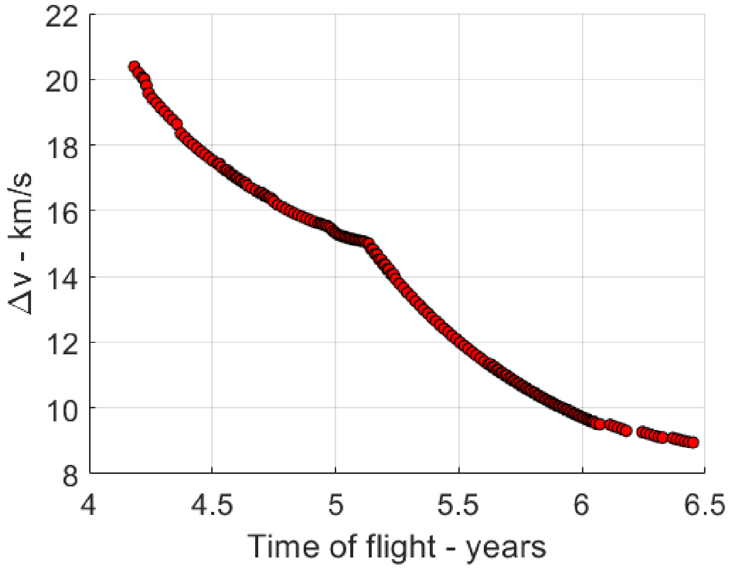


Figure 2.1: Pareto front of Δv and time of flight for a multiple gravity assist mission like JUICE.

Beyond the challenge of finding optimal solutions to mission analysis problems, one is also interested in understanding the flexibility that a given mission has with respect to different parameters, like the launch date, encounter conditions with respect to celestial objects, scientific interest in visiting different celestial targets, maximum Δv allowable for a given manoeuvre, and so on. Therefore, a realistic mission study may be more akin to a Constraint Satisfaction Problem (CSP) in conjunction with the global optimization problem. In the CSP formulation, one wants to find solutions that are compliant with given mission-driven constraints.

A general definition of the CSP is provided in the following [20]. In the CSP one needs to specify a set of variables, e.g., $(\vec{x}(t), \vec{u}(t), t)$ and for each variable a finite set of possible values, namely its domain, and a set of constraints, similarly to any optimization problem. A solution to the CSP is called feasible if an assignment of a value from its domain to every value exists such that all the constraints are satisfied. Therefore, one has:

$$
\begin{aligned}
\text{For a variables set: } & (\vec{x}(t), \vec{u}(t), t) && \text{(states, controls and time)} \\
\text{Subject to: } & \dot{\vec{x}}(t) = f(\vec{x}(t), \vec{u}(t), t) && \text{(dynamics equation)} \\
& \vec{x}(t_0) = \vec{x}_0 && \text{(initial conditions)} \\
& g_i(\vec{x}(t), \vec{u}(t), t) = 0, \forall i = 1, ..., m_{eq} && \text{(equality constraints)} \\
& g_i(\vec{x}(t), \vec{u}(t), t) \geq 0, \forall i = m_{eq} + 1, ..., m_{in} && \text{(inequality constraints)} \\
& \text{Find all, or as many as possible, assignments of a value for each variable such that} \\
& \text{all the constraints are satisfied.}
\end{aligned}
\tag{2.13}
$$

It should be noted that in this case there is no need for an objective function to be explicitly specified. In other words, no minimization/maximization is (necessarily) performed. However, the problem could be easily set up in a way that a solution is found such that some objectives are minimum/maximum (see also Chapter 6).

Regarding the set of assignments to the variables that constitutes the solution of the CSP, one might be interested surely in finding at least one assignment (such that the problem is satisfiable), but one could set up the problem to obtain:

- Just one solution, with no specific preference regarding objective functions.
- All the solutions. This is analogous to an exhaustive search presented in Chapter 1 for the Earth-Mars mission, on which pork-chop plots in Figures 1.2 and 1.3 illustrate all the possible trajectories that satisfy mission-specific constraints.
- An optimal solution with respect to some objective functions (i.e., same as solving a constrained optimization problem like in Eq. 2.8) alongside multiple sub-optimal solutions such that science objectives can be adequately studied and traded-off, as well as the design flexibility can be analyzed (see also Chapter 6).

Following the example of the Earth-Mars mission from Chapter 1, one is clearly interested in finding the minimum spacecraft performances in terms of launch mass for a launch date between two dates. That would be to solve an optimization problem. On the other hand, a constraint satisfaction problem would be to identify the launch window to deliver a given dry mass spacecraft to Mars within the same launch window. As an example, a multi-start (MS) search [110] could be seen as a way of addressing a CSP for mission analysis while looking for the optimal solution. This in fact uses several local minimizations in different points of the search space. A MS search generally employs two steps: (1) the first is determining the starting points for the minimization, (2) the second is the minimization themselves. One could for example discretise the departing date of the Earth-Mars mission and set up an optimization problem to look for the minimum spacecraft mass available for each departing date. Other examples involving continuous-varying variables can be the assessment of the launch declination range to inject a certain mass into spacecraft for a range of dates or the thrust profile that must be below a given maximum thrust threshold. From this example it is clear that the CSP formulation is similar to a feasibility problem that is not an alternative to trajectory optimization, but it rather completes the mission analysis with additional information and trade-off.

In the context of multi-target space missions, a CSP is relevant not only for the continuous-varying parameters, but also regarding the integer part of the problem, i.e., finding feasible sequences of celestial objects to be visited. This is because in preliminary mission design one might be interested in providing sequences of different targets, such that a correct trade off analysis between different sequences considers, for example, the scientific interest of a specific sequence of objects or the encounter conditions with them. This is because addressing the specific science interest of a given sequence of targets is usually a difficult task, and any attempt to define a science function to be maximized would be rather arbitrary and always mission dependent. This is the case of the CASTAway mission proposal [9, 111] and of missions that visit multiple celestial objects in general and are discussed in more detail in Chapter 6.

For the purposes of the present work, the aim is to solve the optimization problem, both in its single- and multi-objective formulation, and the constraint satisfaction problem on the integer component of the mixed-integer design of multi-target mission design.

2.1.3 Design variables

The third building block when designing space missions is to specify the design variables to be employed, that define the specific spacecraft transfer. This step is crucial as it does not solve the problem itself, but rather sets the parameters to solve it. In literature, this building block is also referred to as approaches [14, 25] in the context of spacecraft trajectory optimization.

A first distinction between different approaches is:

- Analytical approaches. These are closed-form solutions to find spacecraft trajectories, that usually work for special cases, not always representing appropriately real-world mission scenarios. An example of analytical approach is the well-known Hohmann transfer [15, 16] for velocity increment minimization between circular-coplanar orbits. Nevertheless, using analytical approaches for orbital transfers is still informative and very useful for a fast preliminary estimation of mission costs and transfer times [112–114].

- Numerical approaches. These become valuable over analytical ones in many practical situations, when the model complexity increases, and closed-form solutions to dynamics equations are no longer available. Among numerical approaches, three methods are identified that are indirect, direct [115], and dynamic programming [116] methods. Main techniques are shooting ones (i.e., single-shooting or multiple-shooting) or collocation.

Indirect methods are based upon calculus of variation that define necessary conditions for an optimal trajectory, i.e. the Euler-Lagrange equations. The dynamics equations are 'adjoined' to the Lagrange term $L(\vec{x}(t),\vec{u}(t),t)$ from Eq. 2.3 by means of the Hamiltonian:

$$H(\vec{p}(t),\vec{x}(t),\vec{u}(t),t) = \vec{p}(t)\cdot\dot{\vec{x}}(t) + L(\vec{x}(t),\vec{u}(t),t) \tag{2.14}$$

where the components of vector $\vec{p}(t)$ are the co-states (or adjoint variables, or Lagrange multipliers) with number equal to the dimension of the state $\vec{x}(t)$. Therefore, the necessary conditions for optimality become:

$$\dot{\vec{p}}(t) = -\frac{\partial H}{\partial \vec{x}} \tag{2.15}$$

with boundary conditions:

$$\vec{p}(t) = \left(\frac{\partial \Phi}{\partial \vec{x}}\right)_{t_f} \tag{2.16}$$

If the final time is to be optimized, the transversality condition should be specified:

$$\left(\frac{\partial \Phi}{\partial t} + \vec{v} \cdot \frac{\partial \Psi}{\partial t} + \left(\frac{\partial \Phi}{\partial \vec{x}} + \vec{v} \cdot \frac{\partial \Psi}{\partial \vec{x}} \right) \cdot \dot{\vec{x}} + L \right)_{t_f} = 0 \qquad (2.17)$$

where $\Psi(\vec{x}(t_f), t_f) = 0$ is a user-defined terminal boundary function. The system in Eq. 2.15 to 2.17 is a two-point boundary-value problem. The optimal control is then chosen according to the Pontryagin minimum principle (PMP) [117] stating that the Hamiltonian should be minimum (or maximum) with respect to each element of the control vector. The main advantage of using indirect methods is that the optimal control is determined by analytical differentiation, and local optimality is assured. However, the dimension of the system increases due to the presence of the co-states dynamics equations and the associated two-point boundary-value problem is generally difficult to be solved without proper initial guesses on the states and co-states, due to the high non-linearity of the system. Moreover, knowledge of variational calculus is required, and the partial derivatives of the Hamiltonian with respect to the control variables might be non-trivial to derive.

On the other hand, direct methods require a parametrization of the problem through a vector of unknown variables that are used as optimizable parameters. The problem is thus converted into a non-linear programming problem [27], on which the objective function is to be optimized with respect to the specified parameters set subject to the constraints. The independent variables (e.g., the time) are discretized and states and controls are computed only at time instants and not in the continuous time domain as for indirect methods. The dynamics equations are enforced by numerical integration between each time instants of the discretization. The main advantages of direct methods are that no differentiation is needed (i.e., no optimal control theory knowledge is required) and constraints are easily included. Nevertheless, the knowledge of the gradients (if available) could potentially be used in non-linear programming algorithms (see also later section 2.1.4) to speed up the optimization process, with respect to, for example, finite differences. The main drawback is that states and controls are only known at discrete points and not continuously.

The third method considered here is dynamic programming that is based upon the Bellman's principle of optimality [116, 118]. According to such principle, the problem is divided into a series of sub-problems that are linked together by a recurrence relation. Although dynamic programming was originally developed for discrete problems (see also Appendix B and later Chapter 5), an equivalent formulation for continuous-time problems has been developed in the form of the Hamilton-Jacobi-Bellman theorem [119]. The main advantage of dynamic programming is that the Bellman's principle of optimality is a necessary and sufficient condition for a solution to be optimal [120], the search space is entirely scanned thus an optimal solution is also a globally optimal solution. However, the main limitation is associated to the exponential increase in the dimension of the state space, an issue commonly known as 'curse of dimensionality' [121]. Some attempts to overcome this issue are found in literature [122–124] that make use of the so-called differential dynamic programming [125, 126]. In this case, the cost function is replaced by a linear-quadratic approximation in the vicinity of a reference solution, thus sacrificing global optimality. The reference solution is usually found on a reduced set of parameters for the problem at hand (e.g., assuming the thrust vector aligned with spacecraft velocity [122] on a low-thrust trajectory design) and employing meta-heuristics as black-box optimizers to solve the reduced problem.

Among the most popular techniques usually involved to solve optimization problems one has either single- or multiple-shooting techniques or collocation. In single-shooting techniques, a guess is made on the initial states and control values (or co-states). The trajectory is then propagated from t_0 to t_f and the constraints on the final states are evaluated at t_f. The process should then determine the initial conditions on states and controls (or co-states) and impose the constraints at the end of the propagation. In multiple-shooting techniques, the interval $[t_0, t_f]$ is divided into N segments to reduce the sensitivity to guesses on the initial conditions. The trajectory is propagated along each interval by guessing initial states and controls (or co-states), that are determined by the optimization process. Continuity between different segments is enforced by means of linear constraints (generally it is a difference between position vectors). The number of constraints is thus significantly higher than in single-shooting techniques. Finally, collocation techniques [127] make use of time grids at which the states and their time derivatives are defined. Quadrature rules or interpolation are then used to find the values of the states and their derivative at intermediate points, which is the main advantage of such techniques. The interpolating function is forced to satisfy the system equations at the center of each interval (i.e., at the so-called collocation points). This forms a constraint set that should be satisfied during the optimization procedure. The main drawback is that the number of controls and constraints depends upon the density of the grid, that can become prohibitive from the computational effort point of view when accurate trajectory representation is required, although the sparsity of the Jacobian matrix can allow large computational speed up.

In the present thesis, a direct approach is used to model interplanetary trajectories. This assumes that the states and controls are replaced by an appropriate set of variables that represent spacecraft trajectories that encounter celestial objects and perform deep space manoeuvres. This model, also referred to as MGA-DSM [128–131], is described in detail in Chapter 3. One of the contributions of this work is the extension of the MGA-DSM model to multiple manoeuvres on each planet-to-planet leg, so that complex transfer scenarios like missions towards Mercury [132, 133] can be planned with the pipeline presented through Chapters 4 and 5.

2.1.4 Optimization strategy

The last ingredient consists in selecting an optimization strategy to achieve a solution to the problem as from section 2.1.2 based on the approach selected in section 2.1.3.

It is important to notice that, as anticipated in Chapter 1, in multi-target mission analysis one has also integer variables that affect the cost of the mission, namely the orbital waypoints to be visited. In this sense, the objective functions and the constraints become also functions of integer variables, i.e., $J = J(X, \vec{x}(t), \vec{u}(t), t)$, X being the vector of the orbital points to be visited. This makes the problem more complex since many different trajectory options exist for a single choice of vector X(as discussed in Chapter 1) that satisfy the constraints and that minimize some objective function.

Therefore, one might want to distinguish between two areas:

- The first one is about the solution of the trajectory for a given sequence X of target points in space.
- The second one is about the solution of the full mission analysis problem. This concerns the selection of X becoming part of the problem itself, and the solution of a trajectory for the given choice of X.

If the sequence X is known, one has typically two main options for achieving a solution, that are non-linear programming (NLP) or meta-heuristics.

NLP is a gradient-based method that allows to compute increments in the vector of design parameters in the direction of the minimum (or maximum) of the objective function. Therefore, being y a generic vector of design parameters such that the objective function can be written as:

$$J = f(y) \tag{2.18}$$

one has the following update rule from iteration k to $k+1$:

$$y_{k+1} = y_k + \sum_{i=1}^{n} \alpha_i \left(p_i + \sum_{j=1}^{n} \alpha_j q_{ij} \right) \tag{2.19}$$

where p_i and q_{ij} are the search directions provided by first- and second-order gradients information, respectively, of both objective function and constraints (n being the dimension of the problem), while α_i and α_j are the steps length. It is not necessary to provide both first and second-order gradients information. For example, only first-order gradients available, one could find the steepest ascent (or descent) direction and a hill-climbing method [134] to assess the optimality. Many commercial NLP solvers are available, among which the most popular are the SNOPT [135], IPOPT [136], KNITRO [137] or WORHP [138]. The main advantages of NLP solvers are their quick convergence (due to the information provided by gradients) and robustness. However, their main disadvantages are that, as stated, gradients information needs to be provided and initial guesses of all the parameters set y are needed, which is not always straightforward, making them 'by nature' local trajectory optimization methods.

The other main branch of solutions' methods is represented by meta-heuristics. The main difference from the NLP solvers is that the solution update rule as from Eq. 2.19 is not given deterministically (i.e., based upon gradient information), but rather stochastically. Typical classifications [28, 139] divide the meta-heuristics in single-solution algorithms, i.e., that improve a single solution iteratively, or population-based algorithms, i.e., that improve a population of solutions (also called individuals) at each iteration. Among single-solution algorithms one finds simulated annealing [140], tabu search [141], iterated local search [142]. Population-based algorithms are also very popular in spacecraft trajectory optimization field, and among them one finds genetic algorithms [143, 144], particle swarm optimization [145, 146], differential evolution [147] and ant colony optimization [148]. A general update rule of the solution vector y from iteration k to $k+1$ is:

$$y_{i,k+1} = \Pi(y_{i,k}) \tag{2.20}$$

with $i = 1, ..., n$ (n being the number of individuals in the population) and $\Pi(y_{i,k})$ being an algorithm-specific probability function of y at iteration k. The main advantage of using meta-heuristics is that they do not require initial guesses on the solution (the initialization on y_i is usually done via a randomization generator) and no information about gradients are needed. This makes meta-heuristics global trajectory optimization methods, as the stochasticity processes from Eq. 2.20 can allow the algorithms to escape local minimum funnels. However, meta-heuristics usually depend upon several user-defined parameters, and there is no guaranty that a specific set up of such parameters (whose selection is usually made by a trial-and-error tuning process) would converge to a global minimum consistently, due to, again, their intrinsic stochastic nature. Moreover, including constraints in the search is not trivial, and typically penalty methods are required [143, 144] and usually only a limited number of them can be efficiently handled.

Regarding MINLP/HOCP problems, a general objective function can be written as:

$$J = f(X, y) \tag{2.21}$$

Where both the target points X and the vector y representing set of variables encoding the trajectory states and controls are part of the design. Two options are usually to be considered for solution construction of such problems that are integrated approaches and multi-level approaches [31].

On the one hand, integrated approaches usually define a mixed-integer non-linear programming (MINLP) problem [46, 47], also known as hybrid optimal control problem (HOCP) [48], on which the objective function as in Eq. 2.21 is optimized with respect to a formulation like Eq. 2.8. The key aspect of integrated approaches is that the choice of the mathematical models, objectives/constraints, and transfer strategies (sections 2.1.1, 2.1.2 and 2.1.3) is made once for all and the whole problem's solution is obtained applying some NLP methods or meta-heuristics, or a combination thereof, as black box. A typical example is nested-loop optimization [149], on which an outer loop selects the targets X to be visited, and then an inner loop finds one or more trajectories (i.e., vector y) that visit the given points in X that meet the mission objectives as from Eq. 2.8. Typically, evolutionary algorithms are employed both as outer loop optimizers and inner-loop optimizers [149–152], while NLP methods are sometimes used for the inner loop optimization [153]. The main advantage of using integrated approaches is that a single model/objectives/constraints/transfers are employed in the design, thus a single optimization is required on a set up that is made of a single fidelity. The primary drawback of such approaches is that optimizing even a single choice of targets X is already a complex global optimization problem on its own, and a variation of a single variable in X results in a completely different set of trajectories. Moreover, in most practical scenarios even the dimension of X, i.e., the number of targets to be visited, is not known and any integrated approach should make use of variable-length vectors in the optimization; particularly, adding even a single target to X might result in a considerably increase in the y vector. Such approaches are thus useful for relatively small problems in terms of X domain (i.e., short sequences of targets), or when the knowledge on y is somewhat assumed (e.g., for interplanetary trajectories, on the launch date, manoeuvres location and so on).

On the other hand, opposed to integrated approaches, there are multi-level approaches. These assess the design of multi-target missions in successive stages on an overall multi-fidelity framework [30]. As anticipated in Chapter 1, multi-level approaches are particularly useful in large search domains as those proposed in GTOCs [62,68,69]. The scope of using multi-level approaches is to quickly assess the feasibility of different sequences of targets X with approximated models and then to refine one (or more) promising trajectories for a fixed choice of X. On a first level of fidelity, the transfer between successive targets in X are thus modelled by means of some approximations to estimate critical mission-related parameters like the propellant consumption or the mission lifetime. Examples include the use of energetic considerations to reach different celestial objects [154], replacement of continuous-thrust arcs with impulsive manoeuvres [17, 155, 156], or even deep-neural networks [72, 157–159]. The problem then becomes very similar to the classic TSP [160], that is a combinatorial problem on which the visiting order of celestial targets is to be decided according to some objective function and/or constraints. The main drawback of multi-level approaches is that the fidelity of any model employed should be taken with care as optimal solutions might be discarded in the approximation stage (see Chapter 4).

A typical way of solving the spaceflight TSP-variant is by modelling the search space as a tree-graph [161] that is made by successive levels, each having an increasing number of nodes representing the targets to be visited. The connections between the nodes, i.e., the so-called branches, define the cost of adding one target to the mission. The process of adding successive levels to the tree is called branching. Branching thus involves the usage of the approximate models/objectives/constraints/transfers as anticipated. A full exploration of the tree computes all the possible branches and store all the sequence of nodes. Depth First (DF) or Breadth First (BF) [162] are typical ways to fully explore a tree-graph. The difference between the two lays on the way they perform the branching: DF explores as far as possible along each branch, while BF explores all nodes at the given tree depth prior to moving on to nodes at the next level.

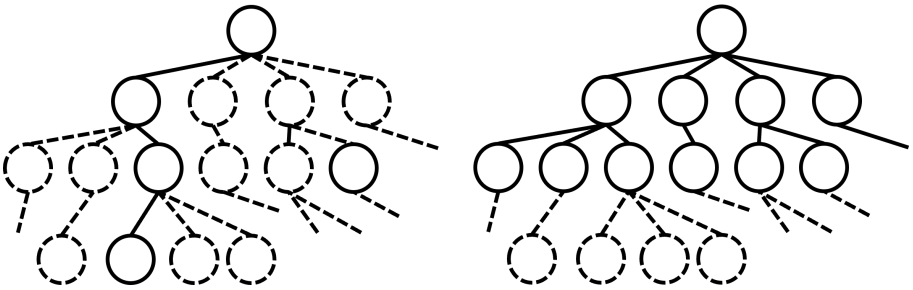


Figure 2.2: Comparison of exhaustive tree-search strategies: DF (left) and BF (right). Dotted nodes are not explored yet.

Figure 2.2 gives a comparison between the two. However, while DF/BF guarantees the global optimality of the solution, these strategies usually become impractical for problems involving visiting many orbital waypoints, as the design space might be too large so that exhaustive searches require an infeasible computational time.

Branch and bound (BB) [163] represents a very useful alternative to obtain globally optimal solutions to the TSP variant of multi-target missions. In BB searches, the cost of reaching any sequence of X targets are progressively bounded with the knowledge of the current best sequence. This defines a threshold, or bound, on the cost function. During the tree expansion, any partial branch that is over the bound is discarded. The major drawback of BB is that if no solutions (i.e., sequences) are found that can lower the bound, BB needs to compute all the possible sequences that compose the search space.

An efficient alternative to avoid enumerating all the possible solutions is dynamic programming [116]. In dynamic programming, if different sequences happen to arrive at the same node for a given tree level, then only one is saved (e.g., the one with the lowest objective function value at the given tree level). This is because any other node attached to the given one would make the cost function vary of a constant value (i.e., the branching cost) for all the available nodes at the next tree level. The main drawback of dynamic programming is that all the possible branches need to be computed to achieve the global optimality and the number of options to be saved at each level of the tree exploration might become too intensive when increasing the number of possible targets to be visited (see also later Chapter 5).

Incomplete strategies are thus relevant that sacrifice the global optimality saving computational effort, i.e., avoiding (1) to expand all the possible branches and (2) to compute all the possible solutions (i.e., sequences) of the problem. Starting from DF and BF, incomplete strategies can be divided in searches that perform the branching either deterministically or stochastically. As paradigms for these two groups, beam search (BS) [44] and ant colony optimization (ACO) [148] are the most famous. In BS the computational effort is bounded by employing heuristics that prevent the exploration of non-promising branches. From Figure 2.3, BS is executed as a variant of BF strategy, on which the exploration of possible trajectory options is performed one depth-level at a time.

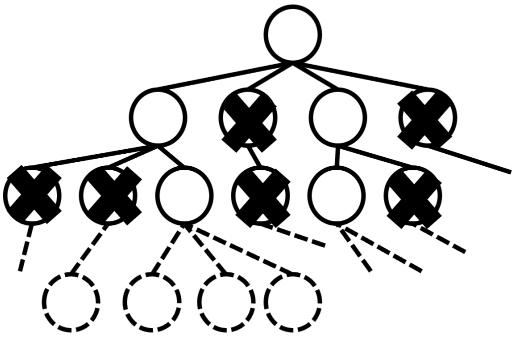


Figure 2.3: Representation of BS algorithm. Dotted nodes are not explored yet, while crossed nodes do not fall within the BW.

From all the branches generated in one level, only a limited set of it, i.e., the beam, is selected to be expanded in successive nodes. The beam selection is performed deterministically, meaning that nodes at each depth-level of the tree are sorted with respect to a heuristic criterion and only those with the highest heuristic are selected for further consideration. The size of the beam is called beam width (BW). BS is one of the most extensively used algorithms to construct solutions for complex multi-target missions like those proposed by GTOCs (see also later section 2.3).

On the other hand, ACO uses stochastic branching based upon a probability function P_{ij} that defines the likelihood to select the node j starting from node i. The probability depends upon: (1) the so-called pheromone τ_{ij}, representing the goodness of solutions computed in previous iteration that contained the branch ij, and (2) the heuristic information η_{ij} that represent the goodness of the branch ij with respect to the other branches available at the current tree level. The P_{ij} is thus given by:

$$P_{ij} = \frac{\tau_{ij}^{\alpha} \eta_{ij}^{\beta}}{\sum_{i=1, i \neq j}^{n} \tau_{ij}^{\alpha} \eta_{ij}^{\beta}} \tag{2.22}$$

where α and β are user-defined weights to τ_{ij} and η_{ij}, respectively. As already discussed, the main advantage of using deterministic strategies is that the set-up of optimization parameters is known that eventually leads to guaranteed global optimality. For example, one can consider the case of the BS, where the tuning of a single parameter is needed, i.e., the BW. The set-up of BS algorithm that leads to the global optimum is known a priori, since having a $BW = \infty$ [2] corresponds to have a DF/BF search that eventually return all the solutions, and thus also the globally optimal one. Being the algorithm deterministic, repeated runs with the same set-up always return the same solutions. On the other hand, when using stochastic meta-heuristics, one does not know a priori the set up that leads to global optimality (or even feasible solutions) consistently. Since the algorithm employs

[2] In this case, ∞ corresponds to select all the available options at each tree-level to be kept for further expansion.

stochastic node selection, repeated runs with the same parameters set-up is not guaranteed to converge to the same solutions. Nevertheless, some applications of ACO to multi-target missions can be found in [57, 98, 164].

Hybrid strategies are also available that make use of a combination of deterministic branching and stochastic meta-heuristics. For example, a possible hybridization between ACO and BS [155, 165] uses the function from Eq. 2.22 to compute the probability to reach any node j from the current tree level (as the standard ACO) and then selects a number BW of them to be kept for further expansion based on their P_{ij} values. In [95], an evolutionary programming algorithm is used to provide lower bound information on the cost function for a deterministic BB to be performed to solve numerous problems in space mission design.

As anticipated in Chapter 1, in the present thesis, a multi-level approach is adopted. On a first fidelity level, the combinatorial problem of visiting different targets is solved by employing deterministic tree-searches based upon dynamic programming principles to guarantee global optimality within limited computational effort. Then, on a successive level of fidelity, candidate solutions are refined by means of global optimization evolutionary algorithms.

2.2 MGA State-of-the-Art

This section focuses on solution approaches employed for MGA trajectory design. One highlights the main gaps in the literature and presents the novelties that are introduced with the present thesis.

In interplanetary missions, MGA transfers make use of successive close passages, also called swing-bys or fly-bys, with planets or other celestial objects to change the spacecraft heliocentric velocity [166] (or planetocentric as in the case of moon tours [167–169]). This permits to gain or lose energy with no propellant expenditure, thus allowing to explore regions in the Solar System that would be otherwise demanding to reach. The design of such missions falls within the multi-target design, as one wants to visit multiple orbital way-points, i.e. planets or planetary moons, to achieve scientific objectives. For example, Galileo [170], Cassini [171] and the more recent BepiColombo [133], Parker Solar Probe [172], Solar Orbiter [173] and JUICE [6] required or will require multiple fly-bys with Venus, Earth or even Jupiter to reach the desired science orbits.

The design of such missions falls within the within the class of MINLP/HOCP problems, as one needs to tackle (1) the combinatorial part, aiming at choosing the optimal sequence of planetary fly-bys, and (2) the continuous part, aiming at identifying one or more locally optimal trajectories for a candidate planetary sequence in terms of planetary phasing, swing-bys and thrust arcs. Following taxonomy presented in section 2.1.4, solution approaches can be either (1) multi-level, or (2) integrated [31]. These are discussed in sections 2.2.1 and 2.2.2 respectively. Section 2.2.3 then summarizes the main gaps and novelties.

2.2.1 Multi-Level MGA Design

As in section 2.1.4, multi-level approaches divide the MGA design problem into several steps. The first one is the generation of the vector X, i.e., the planetary sequence (section 2.2.1.1), while the second one is the generation of optimal trajectories for the sequences identified 2.2.1.2.

2.2.1.1 Tisserand Graphs

A Tisserand graph is a graphical tool which makes uses of energetic consideration to quickly assess the feasibility of different gravity-assist sequences [154, 174]. It has been used over the past decades for trajectory analysis and design of interplanetary missions. Some examples include the Galileo Orbiter's trajectory design [175], as well as Europa Orbiter design [167–169], or even winning trajectory design of the 6th edition of the Global Trajectory Optimization Competition (GTOC) [54]. Several modified versions were developed to adapt it to specific purposes, as the Saturn moon tours [176–178], the case of circular-restricted 3-body problem (CR3BP) [179], and low-thrust propulsion trajectories [180].

The main issue with Tisserand graphs is that they contain no explicit information about planetary phasing and transfer times between consecutive swing-bys. Therefore, any information derived from Tisserand graphs on such critical parameters [154] should be taken with care as it might not correctly map in higher fidelity models [181]. However, useful information about the final MGA trajectory can still be inferred from Tisserand graph analysis, mainly about planetary encounter conditions, MGA sequence length and globally optimal paths in terms of Δv consumption (see also section 4.3.1). Moreover, most of such approaches imply simplified dynamical models for both the swing-by objects and spacecraft motion, assuming circular-coplanar paths for swing-by objects and zero-inclination orbits for spacecraft transfer between two successive encounters.

Although this is a generally good approximation for most interplanetary missions, one can envisage mission options that might benefit from inclined orbits for specific scientific purposes, that are not usually addressed in literature. For example, two of the latest ESA interplanetary missions, namely Solar Orbiter [173] and JUICE [6, 182], aim to achieve high orbital inclination to observe never-explored latitudes in the near-Sun and Jupiter regions, respectively. Moreover, novel mission concepts like Dolphin [75] also explore the possibility of using inclined heliocentric orbits to reveal the nature of interstellar and interplanetary dust particles [75].

2.2.1.2 Optimization of MGA Sequences

If the MGA sequence is known, one needs to find one or more trajectories that visit the selected planets while satisfying constraints and possibly minimising some objective functions.

A very useful option in terms of solution strategies is represented by (deterministic) grid approaches. These make use of systematic scan of the search domain, in terms of launch window and transfer times between consecutive planetary encounters, usually coupled with incremental pruning techniques, to reduce the dimensions of the search space. These have generally been used to design ballistic MGA trajectories when virtually null thrust is required. In fact, approximated Δv manoeuvres at planetary swing-bys are generally assumed and the optimization routine, usually single-objective, looks for trajectories that minimize such Δv to a trivially small number. For example, NASA's STOUR program [183–185], assuming the overall MGA sequence to be known, considers the difference between incoming and outgoing planet-spacecraft relative velocities at the planetary encounter as a measure of the Δv cost for a given planet-to-planet phase. Methods that consider DSMs while in the search for optimal trajectories in STOUR have been studied in [186, 187], resulting in a huge number of trajectories that STOUR had to analyse, thus leading to long computational time required. ESA's GASP program [188], also assuming the knowledge of MGA sequences, employs the so-called powered swing-by model to link successive legs of the overall MGA mission at planetary encounter, assuming a Δv manoeuvre occurring at the moment of closest passage with a planetary swing-by. GASP solutions are then used to inform successive optimization with genetic algorithm, particle swarm optimization or differential evolution. ESA's SOURCE algorithm [189] uses Δv manoeuvres applied immediately after the planetary encounters or Δv of a fixed magnitude for legs involving the same planet for consecutive swing-bys for estimating the cost of a planet-to-planet phase, with the knowledge of planetary sequence. A similar approach was employed by DEIMOS team in the context of GTOC1 problem [190], which is based on a systematic ballistic search of MGA trajectory options with manual selection of best-suited options, fully refined by means of a low-thrust optimizer. Another tool called PAMSIT has been presented in [191] which solves the ballistic MGA trajectory design, also accounting for aero-gravity assist manoeuvres, without knowing a priori the planetary sequence. PAMSIT uses an energy-based criterion to find planetary sequences without accounting for the planetary phasing, that are then used in a successive search for actual trajectories in a simplified circular-coplanar dynamics for the planetary motion. However, having simplified dynamical model can lead to non-precise solutions in a successive higher-fidelity model, as discussed in Chapter 4.

The main attractive feature of deterministic solution strategies is that the set-up that guarantees global optimum solutions is known. However, the described approaches all require approximated Δv manoeuvres at planetary swing-bys that do not necessarily correspond to actual Deep Space Manoeuvres (DSMs) in real-world mission scenario. In order to assess the efficiency of such transcription process, one needs to analyse the relationship between different different manoeuvre models types, that is not generally reported in literature.

Moreover, number of routes to be evaluated and stored usually represents an issue in terms of memory storage. This depends upon the coarseness of employed grids, that is always a compromise between number and quality of solutions (see also Chapter 5). The issue is best described by the following example. Consider a mission from Earth (E) to Mars (M) with intermediate fly-by at Venus (V), i.e., the sequence is EVM. The launch window can happen anytime in a 2-years range. The transfer durations on the EV and VM phases are bounded in $[50, 550]$ days and $[50, 850]$ days, respectively. If one discretizes the departure dates and transfer durations ranges into grids with step size of 2 days each, the total number of bins are 365, 250 and 400, respectively. Therefore, for the EV phase, one needs to store $365 \times 250 = 91250$ routes. At the next VM phase, for each route identified, one has 400 more routes, thus $91250 \times 400 = 36.5$ million routes to be sorted and ranked by objective in selected mission duration bins, in order to obtain optimal solutions. It is clear that adding more planets to the sequence makes the problem cumbersome. Although this can be partially mitigated by using filtering criteria (e.g., on the departing velocity, or Δv at fly-bys), it can be shown (Chapter 5) that the memory issue still represents the crux of such approaches. The issue is amplified if DSMs are considered on each planet-to-planet phase, as these imply an increased number of variables to be described [31].

Therefore one might want to consider the application of stochastic meta-heuristic strategies to solve the issue of finding optimal trajectories for a fixed MGA sequence. In this way, one does not need to manage massive grids, but rather a population of solutions (whose number is fixed by the user) that are progressively improved by means of adaptive processes to manage the exploration of the search space.

For example, in [131], the authors solve the single-objective optimization of different MGA sequences using several meta-heuristics, like differential evolution, particle swarm optimization, genetic algorithm and simulated annealing. Such approaches are stochastic in nature and thus do not guarantee the convergence to optimal solutions. In fact, the authors note that the simple application of such solvers ’is not enough to find good solutions and more elaborated approach is desirable’. Moreover, performances of such solvers usually depend upon a number of user-defined parameters that need to be properly tuned, and such tuning is problem-specific. For example, in [192] the problem of tuning a differential evolution algorithm is assessed on single-objective optimization MGA problems, assuming the knowledge of the sequence and the presence of DSMs on user-defined phases of the transfer.

To mitigate such issues (i.e., memory and stochasticity), an incremental pruning algorithm is presented in [193], in an effort to extend the results obtained with GASP to MGA transfers with DSMs, optimizing with respect to the overall Δv consumption. The MGA problem is decomposed into simpler sub-problems and tackled incrementally. The main limitation of the proposed approach is that user-defined partial objective functions need to be defined at each incremental layer, that are optimized by means of a modified monotonic basin hopping algorithm, also assuming the knowledge of the number of revolutions about the Sun.

2.2.2 Integrated MGA Design

To deal with MGA design with integrated approaches, stochastic meta-heuristic strategies are generally employed.

Moreover, they can require quite intense computational effort to handle the mixed-integer complexity of MGA trajectory design. To mitigate such drawbacks, stochastic meta-heuristic strategies are usually employed on very narrow search spaces in terms of launch dates, transfer times and DSMs location and/or magnitude. Due to the mixed-integer nature of the problem at hand, an approach based of nested loop optimization was presented in [149–152]. In nested optimization, integer GA is used on an outer loop to search for the optimal MGA sequence. The goodness of such sequence is assessed in the inner loop looking for single-objective optimal continuous design variables by means of a combination between particle swarm and differential, with computing time rising steeply with the dimensions of the search space, in the order of multiple days in parallel computing [149]. A similar approach is employed in [153], where a gradient-based single-objective optimization is used for the inner loop strategy, but with no guarantee on a consistent convergence to optimal solutions, which is a typical issue of meta-heuristic strategies. Ant colony optimization has also been employed in [98,164] to construct MGA sequences that are Δv-optimal exploiting DSMs at the apses of planet-to-planet transfer arcs, assuming the knowledge of the departing date and a simplified dynamical model for planetary orbits. ACO has also been used in [47] knowing the number of planetary swing-bys and the position of one DSM on a pre-defined leg of the transfer. The concept of hidden-genes GA has also been exploited in [194, 195], assuming the launch happening anytime in a window spanning on a 30-days range and assuming maximum one DSM per interplanetary leg.

Hybrid approaches can be used that exploit deterministic techniques to check the available planetary swing-bys alongside evolutionary programming to solve the continuous optimization problem. This is the main scheme of EPIC software as presented in [95, 196], where single-objective optimization was used to build ballistic transfer and low-thrust arcs from Earth to Mars, as well as a MGA from Earth to Pluto with maximum two swing-by manoeuvres. To account for manoeuvres along the trajectory, the tool IMAGO [128] implements an MGA model with DSMs together with EPIC, that it is tested on missions to Jupiter assuming maximum three planetary swing-bys. An hybridization between incremental tree-graph exploration and bio-inspired probabilistic algorithm has also been proposed in [197] to find optimal solutions with respect to propellant consumption for missions towards asteroids, Jupiter and Mercury.

Most of the above approaches use single-objective optimization, minimizing the overall Δv consumption. However, beyond the challenge to find the global optimum, practical studies for MGA trajectory design usually require a multi-objective optimization, on which multiple objectives that are competing one to each other need to be optimized at the same time.

For the multi-objective formulation of the MGA problem, primarily stochastic meta-heuristics have been employed. Non-dominated-sorting GA (NSGA-II) has been employed in [198], assuming a maximum of three swing-bys and no DSMs during the transfer. NSGA-II has also been used in [199] in conjunction with parametric spreading, and as the outer loop optimizer in [200] with MBH and sparse non-linear optimizer as inner loop optimizer, with quite intense computational effort spread on massive parallel computations. The concept of hidden-genes GA for multi-objective optimization is used in [201], constraining the bounds of the launch date in a one-month range and assuming maximum three fly-bys for missions towards Mercury and Jupiter, and maximum four fly-bys for missions towards Saturn. A multi-objective variants of both ACO [202] and particle swarm optimization [203] have been employed to optimize MGA transfers with the knowledge of the planetary sequence. An agent-based mimetic algorithm has also been introduced [204] for multi-objective optimization assuming a priori the MGA sequence towards Saturn, and considering no DSMs during the transfer.

2.2.3 Gaps and Novelties on MGA Trajectory Design

From the discussion above, one can highlight the following limitations of existing methodologies for designing MGA transfers. These are here summarized:

1. When considering deterministic approaches, the knowledge of MGA sequences is usually assumed. The presence of DSMs is not generally considered, and when they are included in the search, computing times rise steeply, and intensive parallel computing is required. Moreover, even when approximated Δv models are considered, the relationship between manoeuvres model types is rarely discussed, although it is crucial to understand the goodness of a given approach. If Tisserand-based approaches are used to generate MGA sequences, the information that can be derived from Tisserand graphs is only rarely discussed and the exploitation of three-dimensional graphs is not usually reported. Moreover, typically single-objective optimization is addressed.

2. When considering stochastic or hybrid approaches, the main limitations lie on the uncertainty of convergence to global optimality. To mitigate this, search spaces of narrow dimensions are usually employed, assuming fixed number of swing-bys, small launch windows and constrained transfer times between encounters, or presence and/or location of DSMs. Again, mostly single-objective optimization is tackled. When considering multi-objective optimization, either the knowledge of the sequence is assumed a priori, or small search spaces are considered.

Thus, one identifies that there is a gap in literature in exploiting the advantages of deterministic strategies to handle the multi-objective optimization of MGA trajectories in a robust and efficient manner, while mitigating the main issue of meta-heuristic strategies, primarily related to the dimensions of the search space, computing time and a priori knowledge of the trajectory sequence or DSMs positions. To fill the gaps, the novelties introduced in this thesis with regards to MGA trajectory design are thus the following:

1. Multi-objective optimization of MGA transfers with DSMs is performed on a de-

terministic manner. The MGA trajectory design problem is transcribed into a graph of interconnected nodes that are linked by approximated Δv occurring at each planetary swing-by. To mitigate one of the issues related to such approach (i.e., the high number of routes to be considered), dynamic programming principles [116] are extended to such transfers to handle the multi-objective optimization. In this way, efficiency in automatically exploring the meaningful search space is guaranteed, with the minimum computational effort possible. An energy-based criterion based on Tisserand parameter [154] is employed during the search for trajectories to select achievable planets.

2. When assessing the feasibility of different MGA trajectories by means of Tisserand-based analysis, inclined orbital loci are exploited to build MGA sequences for high-inclination orbits. Such transfers are analysed in the context of practical mission design of the recent Dolphin proposal to the 2022 ESA F/M-class call [77].

3. An analytic procedure based on approximated Δv removal using position constraints is used to assess the relationship between manoeuvres model types and a key issue of deterministic approaches, i.e., the correspondence between the grid optimization and actual DSMs. In this way, the robustness of the process is assured, allowing good representation of any manoeuvre required during the mission. It is shown that such an approach could potentially be included in the search step at the price of relatively small increase of computing time.

4. Very large search spaces in terms of launch widows and transfer times are considered in the optimization of long MGA sequences, without assuming a priori any information of the final solution set, namely on planetary encounter, presence, number and location of DSMs and number of revolutions about the main attracting body.

2.3 Approaches from Literature to Asteroids Tours Trajectory Design

Instances of missions that target multiple asteroids and/or cometary objects also represent a growing trend. Examples of this include, but are not limited to, MANTIS [205], proposed in NASA's Discovery Program [3], CASTAway [9, 10], proposed both in 2018 and 2022 for ESA's M-class call (M5 [206] and M7 [77], respectively), or Lucy [11], launched on October 2021 towards the Trojan asteroids clouds. Moreover, missions that can perform either rendezvous of fly-by with main belt cometary objects [207], such as Castalia mission [111] proposed to ESA's M5, extinct/dormant comets [208] or Centaurs [32, 209] are also of paramount importance for future Solar System exploration. As such, they have received attention from scientific community in the context of ESA Cosmic Vision 2050 for future missions [210]. There is thus a clear scientific interest in exploring such objects, to better understand the composition and evolution of early stages of Solar System.

This section first contextualizes the problem of designing such missions within the current literature (section 2.3.1) and then presents the gaps identified and proposed novelties introduced with the present thesis (section 2.3.2).

2.3.1 Multi-Level Approaches on Asteroid Tours Design

As anticipated, the overarching scheme adopted for designing these missions is the multi-level one [60]. On the overall multi-level strategy, two aspects are important: (1) the transcription process and (2) the solution strategy.

2.3.1.1 Transcription Processes

The transcription process allows to transform the mixed-integer problem into a combinatorial one (i.e., on a TSP variant). Such process plays a key role in the efficiency of proposed searches and quality of the solutions, understanding feasible transfers as

[3] `https://www.nasa.gov/planetarymissions/discovery.html` last accessed May 2022

quickly as possible without losing information on the trajectory shape itself. For example, in [211] the distance between the spacecraft trajectory and all the asteroids in the set was used to identify feasible encounters, assumed to occur at the epoch of closest distance. In [155, 165] an approach based upon linear approximation of spacecraft motion and clustering techniques is employed to estimate transfer times and costs between two consecutive asteroid. Although linear approximations are quite helpful in GTOC-like competitions, as the transfer time between asteroids is usually small compared to the overall mission duration, practical mission studies may suffer of inaccuracies due to increased transfer times, number of revolutions about the Sun and presence of planetary MGA. In [62], the time of flight between two consecutive asteroids is considered fixed and derived from a GA-based optimization. In [61], the asteroids fly-by are assumed to occur at their nodal points, after a pre-pruning step of the whole asteroids dataset by means of Minimum Orbital Interception Distance (MOID) between the orbits of the asteroids and the initial orbit of the spacecraft traversing the belt. In [212], a deep neural network [213, 214] is trained on a massive database of Earth-asteroid-Earth trajectory blocks. This scheme (i.e., machine learning + tree exploration) is receiving deeper attention in recent literature [72, 157–159] and possibly represents interesting future research direction.

In the present work, a MOID-informed transcription is used. In this way, the time at which the asteroids can be visited is fixed by the MOID crossing point with respect to a reference trajectory. Although this is a similar approach of GTOC-related approaches [61], on which nodal points are used, one notices that this modified MOID application is useful if planetary fly-bys are not included in the search, since nodes are not optimal points for asteroids fly-bys when MGA trajectories are considered.

2.3.1.2 Solution Strategies

A solution strategy should then be chosen to tackle the transcribed problem (see also section 2.1.4). This is done by setting up a global optimization problem, similarly to a

TSP variant.

Typically deterministic strategies are employed on such variant of GTOC-like trajectory optimization. In fact, a common practice is to apply BB with pruning criteria (also referred to as branch and prune [61]) to limit the number of sub-trees to be expanded. Such criteria are mission-dependent and are usually related to the Δv cost of given branches. In many GTOC-like applications, the dimension of the tree usually becomes cumbersome, thus heuristics that prevent the expansion of non-promising branches are usually employed, at the price of no guaranty in finding global optimality. This is the scheme of the BS as presented in section 2.1.4, which emerged as standard approach to tackle the combinatorial optimization problems in presented GTOC competitions. BS-based approaches have been employed in the winning solutions of GTOC4 [52], GTOC5 [53], GTOC8 [56] and GTOC11 [59], as well as in the third-ranked solution of GTOC5 [63], the second-ranked solution of GTOC7 [155], as well as in the first-, second- and third-ranked solutions of GTOC11 [62, 215]. Apart from GTOC-related literature, a BB approach has been employed [61] to explore the main asteroid belt with a direct launch from Earth, without considering planetary swing-bys. BS is also used in [212] for designing sample-return trajectories, aiming at passing by multiple asteroids and return back to Earth, allowing one Earth swing-by between two asteroids encounter on an overall Earth-asteroid-Earth strategy. The use of stochastic meta-heuristic strategies have been employed as an alternative to tackle asteroid tours trajectory design. Hidden-genes GA has been employed [64, 65] to mutate chromosomes encoding sequences of asteroids. Integer GA has been used in [9, 10] to construct sequences of asteroids from the main belt once the fly-by epoch is known from MOID-based information, assuming also the presence of a Mars fly-by to reach the main belt. In [216], GA is used to study trajectories to sample return missions with Earth fly-bys considering a maximum of three asteroids to be visited. Particle swarm optimization is used as main optimization block for building transfers in [217], targeting one single asteroid per sequence, also allowing a single Mars gravity assist, as well as in [218], considering one near-Earth asteroid and one main-belt asteroid either with one or two Mars gravity assists, and in [219], considering either Earth or Mars gravity assists to reach maximum two near-Earth objects and the region of the main asteroid belt. An ACO algorithm with backtracking scheme is presented in [220, 221] to overcome the main difficulties of stochastic strategies in solving the combinatorial part of asteroids missions. These approaches are however generally outperformed by tree-based approaches, because of the high non-linearities in the integer domain of such trajectories [220]. Approaches that employ hybrid strategies between deterministic branching and stochastic node selection are also relevant. For example, in [165] a population-based ACO [222–224] is hybridized with a BS [225], on which non-dominated sorting [226] is applied at each depth level of the tree graph to rank branched options. The setup assumes a single root node and does not consider planetary gravity assists. In [62], a nested-loop optimization is employed: single-objective BS is used as inner-loop scheme to build asteroid trajectories without gravity assist manoeuvres, while GA is employed as outer loop algorithm to identify optimal asteroids-to-asteroids transfer times.

In the present thesis, a modified dynamic programming approach is applied. Compared to the presented literature, this guarantees to obtain globally optimal paths on the transcribed space with reasonable computational effort, while obtaining wide sets of trajectory options that are relevant in preliminary mission designs (see also later section 2.3.2).

2.3.2 Gaps and Novelties on Asteroids Missions

From the above discussion, most of the mentioned optimization problems tackle the MINLP problem of asteroid tour trajectories from a global optimization point of view, usually employing the overall Δv consumption as performance parameter. However, beyond the challenge to find the global optimum of a complex MINLP, practical mission feasibility studies for asteroid tour missions, also require a description of the topology of the feasible search space, rather than only the identification of a global optimum. Hence, a realistic mission study may be more akin to a CSP [20] rather than to a global optimization problem. In the CSP formulation, one wants to find (one or more) solutions that are compliant with given mission-driven constraints, and among those an optimal, or at least good, solution with respect to an objective function (see again section 2.1.2). Typical solution strategies to tackle the CSP problem on TSP-like applications are again BB-based [227], usually employing some backtracking scheme [19, 228–230] to correct non-useful expansions in early stages of tree exploration.

In the context of asteroids tours, a CSP is interesting because the range of different possible asteroid tours is relevant for scientists within an indicative spacecraft design limit (e.g., Δv related). Moreover, in the case of an asteroid tour mission such as CASTAway, obtaining the CSP solutions is based on a set of integer variables describing the asteroid tour all of which result in a broadly similar spacecraft operational environment. Therefore, there are clearly two needs that should be addressed when designing such missions. The first one is to find the actual global optimum of the MINLP problem at hand. The second one is to solve a CSP to provide adequate extent of mission design options usually required in preliminary design. It should be noted that for the purposes of the present thesis the CSP is tackled on the combinatorial problem (i.e., on the TSP-variant).

From the discussion above, one can highlight the following gaps in existing methodologies for designing multiple asteroid tours. These are here summarized:

- When addressing asteroid related MINLP/HOCP problems with long ($>$ 5 asteroids) sequences of encountered objects, usually the use of MGA trajectories is not considered. For example, in all of the GTOC-related competitions either no gravity assist manoeuvres were allowed [211,231–236] or only the Earth was considered as available planet [237]. This is mainly because of the intrinsic challenges posed by such missions. When MGA trajectories are taken into consideration, usually short asteroid sequences are considered, or only the accessibility of the main-asteroid belt region is assessed. Moreover, limited number of gravity assist strategies are considered, usually employing Earth and Mars, while strategies with Venus and Earth fly-bys are also competitive.

- Usually a single-optimization problem is formulated when designing mission that visit multiple asteroid, generally with Δv as cost function, or with some hybrid cost functions encoding some mission-specific criteria (this is the general scheme of GTOC competitions).

- Linked to the previous point, combinatorial problem formulation of different asteroid sequences as tackled in literature does not consider the exploration of meaningful search space as a fitness criterion (in a CSP-like manner), while practical mission studies should aim for providing as many trajectory options as possible, to maximize the extent of transfer opportunities from a scientific point of view, thus appropriately informing preliminary designs.

To fill the aforementioned gaps, the following novelties are introduced with the present thesis:

- Inspired by the challenges posed by asteroid exploration missions such as CASTAway [9, 10] and MANTIS [205], MGA trajectories are considered while looking for long sequences of asteroid fly-bys, on larger dataset of asteroids compared to GTOC-related competitions and other mission design studies. The study has supported CASTAway proposal to the 2022 ESA's M-class mission call, enabling a comprehensive main-belt exploration opportunities for 2030-2040 launch window.

- Similarly to section 2.2, a modified dynamic programming approach has been developed to efficiently build long asteroids sequences alongside MGA manoeuvres. The concept of dynamic programming is here extended to account for multiple solutions with the aim of maximizing the number of visitable asteroid when exploring the search space, due to the scientific interest of having multiple asteroids tours in preliminary design phases.

Chapter 3

Modelling Interplanetary MGA Trajectories

This Chapter presents the mathematical framework used to design interplanetary missions in the present thesis. The function of this Chapter is to provide fundamental background theory to understand the basic models used in the thesis. Such models, as explained in Chapter 1, define the dynamics of the spacecraft and describe the set of variables that characterize interplanetary multi-target missions with deep space manoeuvres (DSMs).

The Chapter is divided as follows: section 3.1 introduces the mathematical models and assumptions used in the context of the present thesis, namely the patched-conics and linked-conics. These are useful to model critical mission phases like the launch of an interplanetary probe and the fly-by with celestial objects. Section 3.2 describes the main manoeuvres that modify the interplanetary trajectory, i.e., fly-bys and DSMs. Section 3.2 then wraps the content of the previous section and defines the main steps that are needed to find trajectories in the context of MGA missions with DSMs. Section 3.4 presents the conclusions.

These models, i.e., linked-conics MGA with linked conics and DSMs, are encoded in ASTRA toolbox (see Chapter 1) and used in successive Chapter 4, 5 and 6 to compute trajectories (namely, spacecraft position and velocity history with respect to time) as part of the multi-target mission design process.

3.1 Patched-Conics and Linked-Conics

A very useful model employed in the preliminary design of interplanetary MGA trajectories is the so-called patched-conics approximation [16]. This assumes that the motion of a point with negligible mass that is subject to the gravitational action of different bodies can be described by the union of a number of arcs on which only one gravitational action is active, and the others are null. The patched conic assumption works well for MGA trajectories, as it is able to reproduce a wide range of missions [16,25], and it can be shown that it is sufficiently accurate for preliminary design of interplanetary missions [16]. In the patched-conic approximation, the massless point represents the spacecraft, moving under the attraction of a main body, like the Sun for interplanetary trajectories in the Solar System, or planets, and it performs fly-bys with celestial objects (like other planets, moons, or asteroids) every time it flies within their sphere of influence (SOI) [16]. This is a region centered on an object (e.g., planet) on which the gravitational force of the object prevails those of all the other bodies. Thus, within the SOI, the spacecraft motion is assumed to be affected by the planet gravity only. The radius of the SOI is given by [16]:

$$r_{SOI} \approx a(\frac{m}{M})^{2/5} \tag{3.1}$$

where a and m are semi-major axis and mass of the smaller object (e.g., planet), and M is the mass of the main attracting object (e.g., the Sun).

The arcs are solutions of a set of differential equations that could be generally written as [16] (similarly to Eq. 2.2):

$$\ddot{\vec{r}} = -\frac{\mu}{r^3}\vec{r} \tag{3.2}$$

Eq. 3.2 describes the well-known two-body dynamics, for which a closed-form analytical solution exists, and defines the motion of a point that is moving under the acceleration due to the gravity field of a main attracting body with gravitational parameter $\mu = GM$ (M being the mass of the main body). In Eq. 3.2, $\vec{r}$ is the position vector of the point with respect to the active gravitational body, $\ddot{\vec{r}}$ is its second time derivative (i.e., the acceleration). It can be shown that solutions to Eq. 3.2 are conic sections (i.e., circles, ellipses, hyperbolas, and parabolas). Eq. 3.2 describes an ideal situation in which only two bodies (a massless point and a main attractor) interact due to the gravitational force between them, and no other forces affect the motion of the point. In higher fidelity models, the presence of other bodies (like planets, asteroids, comets, moons) modify the path of the point through some perturbative effects.

Let's consider again the Earth-Mars example from Chapter 1. In patched conics approximation, the spacecraft is launched from the Earth and travels on a hyperbolic path with respect to the planet until it reaches the edges of the SOI of the Earth. When inside the SOI, the spacecraft is too close to the Earth to be considerably affected by other gravitational forces, thus only the Earth's gravity is accounted. Once at the edge of the SOI, the gravity pull of the Earth is too small to have a significant effect if compared with the one of the Sun, thus the Earth gravity effect is neglected and only the one of the Sun is considered along the travel towards Mars. The spacecraft finally arrives at the edge of the SOI of Mars, where the Mars gravity is 'switched on' and the one of the Sun is 'switched off'. Analogously to the launch phase, the spacecraft then follows a hyperbolic path with respect to Mars to perform, for example, an orbit insertion around the planet, or a fly-by.

A limit case of the patched conic model is represented by the so-called linked-conics model. Under this approximation, the SOI of gravity-assist bodies degenerates to a point in space. This is a good approximation if the mass of the main attracting body is several orders of magnitude bigger than the masses of the other bodies, which is not far from reality for most practical cases (e.g., MGA trajectories with planetary fly-bys in the Solar System, with the Sun as main attracting body). The added simplification is that the point at which the spacecraft enters the SOI of the gravity-assist body is not defined. Therefore, in order to encounter an object in space, the spacecraft only needs to match its position with that of the object. The linked-conics model is the one employed in this work as it is deemed sufficient to represent a wide range of mission options. Major violations to such models could be experienced when larger planets are considered, like Jupiter or Saturn, for which the gravity effects become relevant way before the planets' positions are matched, or if the spacecraft relative speed to the target planet is low (e.g., as a result of a manoeuvre/low-thrust arc to decelerate). In other words, the accuracy of the model reduces with respect to the distance to the fly-by planet and the planet-spacecraft relative velocity, or infinity velocity, such that inaccuracies arise when the probe is close to the planet and relatively slow (e.g., about 1 or 2 km/s). However, these are in general small and can be corrected in successive stages of the design of a space mission. Such models have also limited accuracy when the spacecraft travels within strong gravity fields (e.g., the one of Jupiter or Saturn) in moon tours, as non-crossing orbits for moon capture (e.g., with Ganymede) [25] are not identified.

Therefore, the arcs resulting from Eq. 3.2 can be further classified in two additional types [31]:

- coast (or ballistic) arcs, along which the spacecraft is subject only to natural forces (like the gravity or other perturbative effects like the solar pressure)

- thrust arcs, along which the spacecraft is also subject to an action coming from the propulsion system .

In the present thesis, only coast arcs are relevant, as different arcs can be connected successively by means of instantaneous events that may represent the effect of a fly-by with a celestial object or the action of the propulsive system through DSMs.

3.2 Events along the Trajectory

Following the discussion from [31], events that are very short in time with respect to the duration of the overall trajectory can be approximated as instantaneous, i.e., occurring at a given time instant and having zero duration. The main implication is that, as an approximation, only the spacecraft velocity (magnitude and direction) is affected by such events, while its position remains unchanged. Examples of instantaneous events that are relevant for MGA trajectory design are the launch of an interplanetary probe, the fly-by, rendezvous, orbital corrections or orbit insertion.

In the following, two main instantaneous events are considered since these are the main ones used in the present thesis:

- fly-bys, which exploit the gravity field of celestial objects to change the spacecraft velocity, and the launch from the departing body
- DSMs that make use of a high-thrust propulsive system to rapidly achieve the required velocity change

3.2.1 Fly-by and Launch

When a spacecraft passes close to a body (e.g., a planet), i.e., within its SOI, the effect of such body's gravity is to modify the motion of the spacecraft. This event can be considered instantaneous in the patched-conics approximation because the spacecraft travels at high speeds with respect to the fly -by body. Therefore, the time spent within the SOI is relatively small compared to the overall interplanetary transfer (in the order of a few days compared to years of travel). Thus, the variation in the spacecraft/object position with respect to the main attracting body (e.g., the Sun) can be considered null, and only the spacecraft velocity is affected by the close passage to the given body.

It can be demonstrated, as shown in Chapter 4, that a fly-by does not modify the magnitude of the spacecraft velocity vector relative to the fly-by body (i.e., the so-called infinity velocity $\vec{v}_\infty$), but only its orientation. The effect of a fly-by with a celestial object (e.g., a planet) is depicted in Figure 3.1. The vector diagram in Figure 3.1 represents the effect of a close passage with a generic planet with velocity $\vec{v}_{pl}$. The spacecraft velocities before and after the fly-by are $\vec{v}^-$ and $\vec{v}^+$, respectively. The turning angle δ determines the orbit of the spacecraft after the fly-by , rotating the infinity velocity vector from $\vec{v}_\infty^-$ to $\vec{v}_\infty^+$. The angle α measures the direction of $\vec{v}_\infty$ with respect to $\vec{v}_{pl}$ and it is useful to graphically map the effect of the fly-by for different v_∞ as done in Chapter 4.

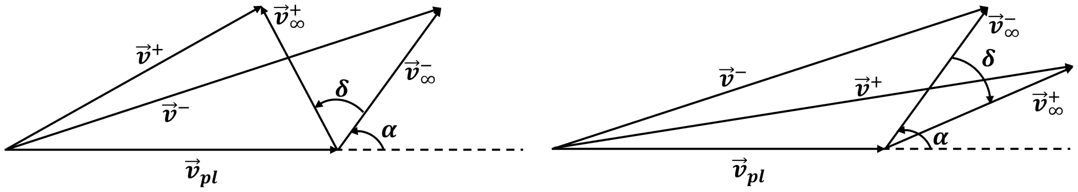


Figure 3.1: Vector diagram representing the effect of the fly-by in front (left image) and behind (right image) a given gravity assist body. Left and right cases result in minimizing and maximizing the energy after the swing-by, respectively.

The turning angle δ, governing the fly-by, depends upon the gravity of the fly-by body (via the gravitational parameter μ), the minimum distance of the spacecraft to the fly-by body (i.e., the periapsis r_p of the hyperbolic passage) and the infinity velocity magnitude v_∞. The relation between δ, μ, r_p and v_∞ is the following:

$$\sin\left(\frac{\delta}{2}\right) = \left(1 + \frac{r_p v_\infty^2}{\mu}\right)^{-1} \tag{3.3}$$

Specifically, the bending increases as close the spacecraft passes to the body and as much the body is massive. On the other hand, the effect of the fly-by decreases as fast the spacecraft passes with respect to the planet. Figure 3.2 represents the geometry of the fly-by and illustrates the relation between δ, r_p and v_∞. The angle δ ultimately corresponds to the angle between the asymptotes of the fly-by hyperbola.

Since this work employs the linked-conics approximation, the point at the edge of the SOI of the planet (i.e., at the 'infinity' with respect to the planet) is not defined, since in the heliocentric reference frame the SOI has null radius. This implies that there is a

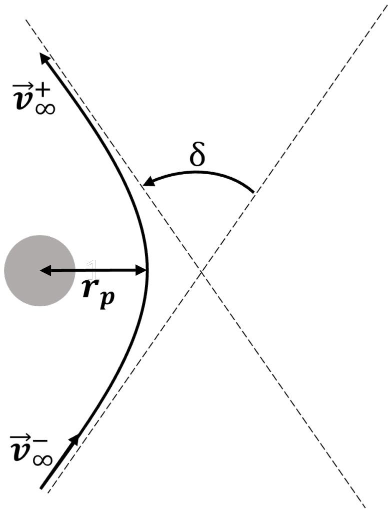


Figure 3.2: Geometry of a fly-by.

degree of freedom that is left free. This is the angle describing the orientation of the fly-by plane, which is the plane that contains the hyperbola, i.e., the one containing the incoming relative velocity vector and the centre of mass of the planet.

To define the plane attitude, the choice of the additional angular parameter for the fly-by is rather arbitrary. In this work, we employ the angle ζ defined as [131]:

$$\vec{v}_\infty^+ = v_\infty \left(\cos(\delta)\hat{b}_1 + \sin(\zeta)\sin(\delta)\hat{b}_2 + \cos(\zeta)\sin(\delta)\hat{b}_3\right) \tag{3.4}$$

where:

$$\begin{cases} \hat{b}_1 = \frac{\vec{v}_\infty^-}{v_\infty} \\ \hat{b}_2 = \frac{\hat{b}_1 \times \vec{r}_{pl}}{|\hat{b}_1 \times \vec{r}_{pl}|} \\ \hat{b}_3 = \hat{b}_1 \times \hat{b}_2 \end{cases} \tag{3.5}$$

where $\vec{r}_{pl}$ is the heliocentric position vector of the planet at the time of fly-by. From Figure 3.3 angle ζ is the inclination between the plane generated by $\vec{v}_\infty^-$ and $\vec{v}_\infty^+$ and the plane generated by $\hat{b}_3$ and $\hat{b}_1$.

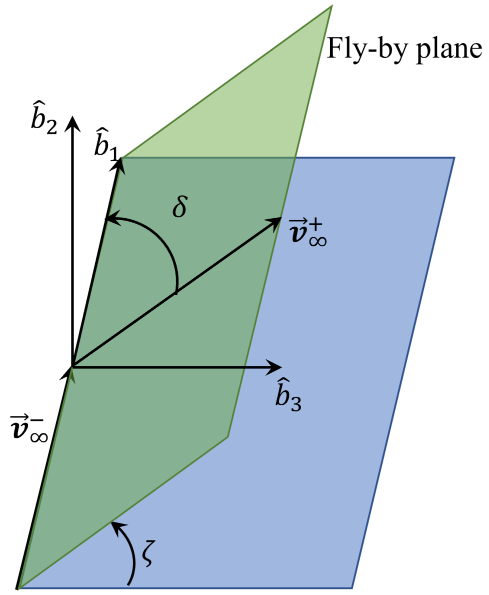


Figure 3.3: Three-dimensional geometry of a fly-by.

To sum up, for a given incoming relative velocity vector ($\vec{v}_\infty^-$) at a planet in position $\vec{r}_{pl}$, the parameters that define the fly-by, i.e., those that allow computing $\vec{v}_\infty^+$ in the heliocentric reference frame, are the periapsis of the fly-by hyperbola (r_p) and the fly-by plane angle (ζ). In fact, through the periapsis r_p one uses Eq. 3.3 to compute the angle δ (having $v_\infty = |\vec{v}_\infty^-|$ and the gravitational parameter of the planet μ), and then one employs Eq. 3.4 and 3.5 to compute $\vec{v}_\infty^+$.

The same linked-conics approximation can be used to model the launch of the spacecraft from a body . The spacecraft is assumed to be injected into a hyperbolic path with respect to the departing planet and leaves the SOI with a given relative velocity magnitude $v_{\infty,dep}$ and directions θ and ϕ that are defined as follows [131]:

$$\begin{cases} \theta = 2\pi u \\ \phi = \arccos(2\nu - 1) - \pi/2 \\ \vec{v}_{\infty,dep} = v_{\infty,dep}(\cos(\theta)\cos(\phi)\hat{\imath} + \sin(\theta)\cos(\phi)\hat{\jmath} + \sin(\phi)\hat{k}) \end{cases} \tag{3.6}$$

where $u, \nu \in [0,1]$ define the heliocentric direction of the departing hyperbolic velocity $\vec{v}_{\infty,dep}$ and $(\hat{\imath}, \hat{\jmath}, \hat{k})$ are defined as:

$$\begin{cases} \hat{\imath} = \frac{\vec{v}_{pl}}{v_{pl}} \\ \hat{\jmath} = \frac{\vec{r}_{pl} \times \vec{v}_{pl}}{\vec{r}_{pl} \times \vec{v}_{pl}} \\ \hat{k} = \hat{\imath} \times \hat{k} \end{cases} \tag{3.7}$$

Therefore, to model the launch of a spacecraft, the three parameters $v_{\infty,dep}$, u and ν are needed, through which the departing relative velocity vector $\vec{v}_{\infty,dep}$ can be computed using Eq. 3.6 and 3.7.

3.2.2 Deep Space Manoeuvres

A DSM is a change in the velocity vector of the spacecraft that is achieved by switching on the spacecraft propulsion system. If the thrust is sufficiently high, then the propellant burning time is relatively short to achieve the required velocity change. In this sense, a DSM can be considered as an instantaneous event, and the related velocity change $\Delta\vec{v}$ is:

$$\Delta\vec{v} = \vec{v}^{+} - \vec{v}^{-} \tag{3.8}$$

where $\vec{v}^{-}$ and $\vec{v}^{+}$ are the spacecraft velocity vectors before and after the manoeuvre, respectively. The magnitude of the manoeuvre is simply the modulus $\Delta\vec{v} = |\Delta\vec{v}|$ that is useful to compute the propellant needed to perform the manoeuvre via the well-known rocket equation:

$$m^{+} = m^{-} e^{-\Delta v / v_e} \tag{3.9}$$

where m^{-} and m^{+} are the spacecraft mass before and after the manoeuvre, respectively, and v_e is the exhaust velocity of the propellant mass. Therefore, minimizing the overall change in velocity Δv required by the spacecraft during its mission corresponds to minimizing the propellant consumption, and consequently the overall mass of the spacecraft.

The approximation of impulsive manoeuvres is sufficiently accurate if high-thrust propulsive systems are considered. If low thrust has to be considered, the equations of motion from Eq. 3.2 should be completed with the acceleration provided by the engine and the mass-flow rate of the propellant, as in Eq. 2.2. However, the impulsive approximation can still be a very informative model even in the cases on which the spacecraft is required to use low-thrust, since one can approximate low-thrust arcs by a succession of coast arcs linked together by DSMs if the acceleration required (i.e., the Δv over the transfer time) is sustainable by the low-thrust engine [155]. This is similar to the so-called Sims-Flanagan transcription [17], which has proven very effective in representing MGA trajectories with low-thrust as first guess [238].

3.3 Gravity Assist Models with Deep Space Manoeuvres

The MGA trajectory with DSMs can be seen as the union of different bits that are (1) the launch, (2) coast arcs, (3) DSMs that link coast arcs on the same leg and (4) fly-bys with celestial objects that link different legs.

Different models are used in the present thesis for modelling MGA trajectories with DSMs, depending upon the number and position of the DSMs. In particular, section 3.3.1 describes a model where each DSM on a planet-to-planet leg is applied at each fly-by encounter; section 3.3.2 then describes MGA transfers with one single DSM applied in between a planet-to-planet leg; section 3.3.3 finally expands the single-DSM model by considering multiple manoeuvres on a planet-to-planet leg.

3.3.1 MGA with Manoeuvres Applied at Planetary Encounters

The idea is to consider for any given planet-to-planet leg a DSM occurring right at each planetary encounter. In this way, each planet-to-planet leg can be modelled as a Lambert arc linking two successive planetary encounters. The cost of the given leg is the velocity discontinuity Δv occurring at the fly-by epoch between incoming and outgoing spacecraft relative velocities with respect to the planet, which are solutions of Lambert problem for the given leg. These Δv can be considered as infinity velocity defects, which represent impulsive manoeuvres applied after each planetary encounter. The defects are computed as:

$$\Delta v = \begin{cases} ||\vec{v}_\infty^+| - |\vec{v}_\infty^-|| & \text{if } \delta \leq \delta_{max} \\ \sqrt{|\vec{v}_\infty^+|^2 + |\vec{v}_\infty^-|^2 - 2|\vec{v}_\infty^+||\vec{v}_\infty^-|\cos(\delta_{max} - \delta)} & \text{otherwise} \end{cases} \tag{3.10}$$

where $\vec{v}_\infty^-$ and $\vec{v}_\infty^+$ are again the spacecraft velocities relative to the swing-by planet before and after the encounter, respectively; δ is the angle between $\vec{v}_\infty^-$ and $\vec{v}_\infty^+$ (positive in the 180 deg-range counter-clockwise) and represents the change of direction between the incoming and outgoing legs of the fly-by; δ_{max} is the maximum possible deflection at the fly-by for the incoming relative velocity $\vec{v}_\infty^-$ and, using Eq. 3.3, reads as:

$$\delta_{max} = 2\arcsin\left(\left(1 + \frac{r_{p,min}|\vec{v}_\infty^-|^2}{\mu_{pl}}\right)^{-1}\right) \tag{3.11}$$

with $r_{p,min}$ being the minimum periapsis of the fly-by hyperbola as in [154] and μ_{pl} the gravitational constant of the fly-by planet.

It should be noted that $\vec{v}_\infty^-$ and $\vec{v}_\infty^+$ are solutions of the Lambert problem between two consecutive swing-bys for a given time of flight T_i, and they have different direction and magnitude. Thus, the overall Δv consumption accumulated along MGA mission ultimately depends on the ephemerides of the objects, through their heliocentric velocities at the encounter epochs which define $\vec{v}_\infty^-$ and $\vec{v}_\infty^+$. In this way, the y vector from Eq. 5.1 and

Table 5.1 only encodes the departing date from the first object (t_1) and the transfer times between successive objects (T_i), i.e., $y = [t_0, T_1, ..., T_{n_{int}-1}]$, and the dimension of the problem is thus $d = n_{int}$ planets in a sequence X. For example, assume the spacecraft needs to follow a Cassini-like sequence, launching from Earth (E) and arriving at Saturn (S), after flying-by Venus (V) twice, the Earth again and Jupiter (J), on an overall EVVEJS sequence, like the Cassini mission [171]. In this case, $d = n_{int} = 5$.

The main advantage of modeling the problem using defects is that each planet-to-planet leg depends only upon the previously visited object through the vector $\vec{v}_\infty^-$. In this way, one exploits the underling graph structure of the search space (see [239] and Chapter 5) to usefully apply graph-traversing techniques based on dynamic programming, allowing efficient exploration of the search space [239]. In this way, one approximates any manoeuvre required during the transfer without much loss of search space information of more complex models (see later sections 3.3.2 and 3.3.3). By employing a dynamic programming-based optimization as described in [239] and in Chapter 5, one has the following solution vector for a Cassini-like EVVEJS trajectory:

$$y = [-769.5, 211, 374, 50, 500, 1252] \tag{3.12}$$

where the departing date t_0 is in MJD2000 and the time of flights T_i are in days. The corresponding trajectory is depicted in Figure 3.4.

One should note that such a solution in the Δv defect model is also a solution in more complex models (as those described in later section 3.3.2 and 3.3.3), and therefore a simple refinement process allows to reoptimize a sequence into the full MGA-DSM dynamical model (as from later section 3.3.2), which may obtain at most some Δv reduction (generally small) [239].

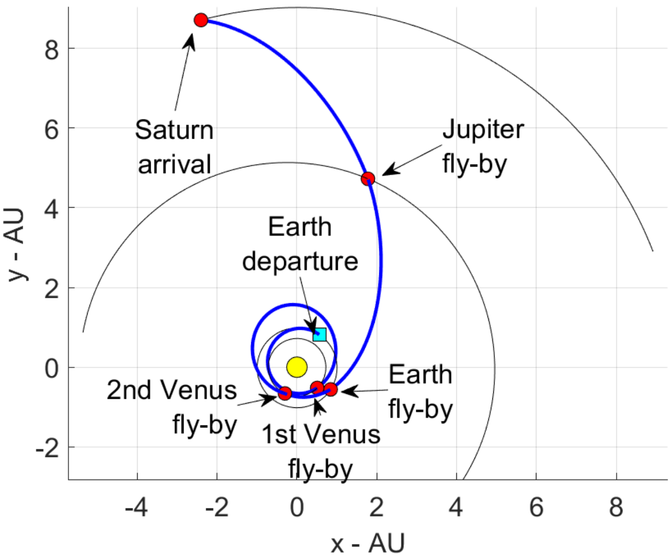


Figure 3.4: EVVEJS Cassini-like transfer within the Δv-defects model.

3.3.2 MGA with one Deep Space Manoeuvre

A very useful model to describe MGA trajectories with DSMs is the so-called MGA-DSM model [128–131]. Solutions to this problem are helpful to preliminary assess real space missions. The MGA-DSM model assumes that only one DSM can occur in each interplanetary leg. The manoeuvre is assumed to occur at a fraction $\eta \in [0,1]$ of the transfer time T within a given leg. Therefore, the variables that describe the overall trajectory MGA trajectory with n_{int} planets in a sequence X can be encoded in a vector as:

$$\begin{aligned} y =& [t_0, v_{\infty,dep}, u, v, T_1, \eta_1, \\ & r_{p,2}, \zeta_2, T_2, \eta_2 \\ & \dots \\ & r_{p,n_{int}-1}, \zeta_{n_{int}-1}, T_{n_{int}-1}, \eta_{n_{int}-1}] \end{aligned} \tag{3.13}$$

where t_0 is the launch date. The dimension of the problem is then $d = 6 + 4(n_{int} - 2)$. Moving the EVVEJS Cassini-like example forward, $n_{int} = 5$, and thus $d = 6 + 4(6 - 2) = 22$ variables are needed to fully describe the trajectory in the MGA-DSM model, which are:

- The launch epoch (t_0)
- The launch event ($v_{\infty,dep}, u, v$)
- The transfer time on each planet-to-planet leg ($T_1, T_2, ..., T_{n_{int}-1}$) and the fraction at which the DSM is performed ($\eta_1, \eta_2, ..., \eta_{n_{int}-1}$). In this example, the number of legs is $n_{int} - 1 = 6 - 1 = 5$.
- The fly-by parameters ($[r_{p,2}, \zeta_2], ..., [r_{p,n_{int}-1}, \zeta_{n_{int}-1}]$) for each planetary encounter before arriving at Saturn. In this example, the number of fly-bys is $n_{int} - 2 = 6 - 2 = 4$.

In particular, the steps to compute spacecraft position and velocity with respect to time in the MGA-DSM model are the following:

i. The departing planet position and velocity in the heliocentric reference frame (e.g., ecliptic-J2000) are known at the launch epoch t_0 through the ephemerides [240], say $[\vec{r}_{pl,1}(t_0), \vec{v}_{pl,1}(t_0)]$. The spacecraft position at the departure corresponds to the one of the departing planet in the linked-conics approximation, i.e., $\vec{r}(t_0) = \vec{r}_{pl,1}(t_0)$. The spacecraft velocity $\vec{v}$ is known from $v_{\infty,dep}, u, v$ by applying Eq. 3.6 and 3.7, thus finding $\vec{v}_{\infty,dep}$. Then, from the velocity triangle as from Figure 3.1, $\vec{v}(t_0) = \vec{v}_{pl,1}(t_0) + \vec{v}_{\infty,dep}$.

ii. The spacecraft initial state is then (analytically) propagated by integrating Eq. 3.2 from t_0 to $t_0 + \eta_1 T_1$, i.e., until the instant at which the first DSM is performed. At this point, the spacecraft has state $[\vec{r}(t_0 + \eta_1 T_1), \vec{v}^-(t_0 + \eta_1 T_1)]$. The superscript '-' indicates again that the velocity is intended before the DSM.

iii. A Lambert arc [38] (orbital two-points boundary value problem) is then used to compute the velocity $\vec{v}^+(t_0 + \eta_1 T_1)$ that brings the spacecraft from $\vec{r}(t_0 + \eta_1 T_1)$ to the position of the next planet in the sequence that is $\vec{r}_{pl,2}(t_0 + T_1)$ in the transfer time $T_1(1 - \eta_1)$. The planet state $[\vec{r}_{pl,2}(t_0 + T_1), \vec{v}_{pl,2}(t_0 + T_1)]$ is again known from ephemerides. A DSM is thus computed as $\vec{v}^-(t_0 + \eta_1 T_1) - \vec{v}^+(t_0 + \eta_1 T_1)$. The arrival state of the spacecraft at the planet is $[\vec{r}(t_0+T_1), \vec{v}^-(t_0+T_1)]$, where $\vec{r}(t_0+T_1) = \vec{r}_{pl,2}(t_0+T_1)$.

iv. At the encounter with the next planet, the incoming relative velocity $\vec{v}_\infty^- = \vec{v}^-(t_0+T_1) - \vec{v}_{pl,2}(t_0+T_1)$ is computed and by means of $r_{p,2}, \zeta_2$ and $\vec{v}_\infty^+$ is found using Eq. 3.5 and 3.5. The state of the spacecraft after the fly-by is then $[\vec{r}(t_0+T_1), \vec{v}^+(t_0+T_1)]$, where $\vec{v}^+(t_0+T_1) = \vec{v}_{pl,2}(t_0+T_1) + \vec{v}_\infty^+$.

v. For each successive leg in the transfer, steps (ii) to (iv) are repeated until the last planet is reached.

As a simplification, one could consider a different way of departure, if the first leg of the MGA trajectory has no DSM. In this case, a simple Lambert arc between $\vec{r}_{pl,1}(t_0)$ and $\vec{r}_{pl,2}(t_0+T_1)$ in the transfer time T_1 is solved (thus $\vec{v}(t_0)$ is computed) and the departing infinity velocity is $\vec{v}_{\infty,dep} = \vec{v}(t_0) - \vec{v}_{pl,1}(t_0)$. This allows a reduction of the number of variables needed to describe the MGA trajectory, which are now:

$$\begin{aligned} y = [&t_0, T_1, \\ &r_{p,2}, \zeta_2, T_2, \eta_2 \\ &\ldots \\ &r_{p,n_{int}-1}, \zeta_{n_{int}-1}, T_{n_{int}-1}, \eta_{n_{int}-1}] \end{aligned} \tag{3.14}$$

The dimension of y is now $d = 2 + 4(n_{int} - 2)$. In the example of EVVEJS, $d = 18$ if no DSM are included in the first leg. The solution vector for Cassini-like EVVEJS trajectory [171] (no DSM on the first leg) is:

$$
\begin{aligned}
y =[&-806.4, 193.4,\\
&1.334.1, 4.659, 420.9, 0.4926\\
&1.219, 4.295, 55.67, 0.02451,\\
&1.082, 4.739, 502.6, 0.02429,\\
&160.9, 4.717, 1255, 0.04327]
\end{aligned}
\tag{3.15}
$$

The departing date t_0 is in MJD2000, the time of flights T_i are in days, the periapsis of the fly-by hyperbola $r_{(p,i)}$ are divided by the radius of the corresponding fly-by planet, and the fly-by angles ζ_i are in radians ($\forall i = 1, ..., n_{int} - 1$). This vector is the result of the application of dynamic programming optimization on the Δv defects model and successive refinement in MGA-DSM model as from [239] and Chapter 5.

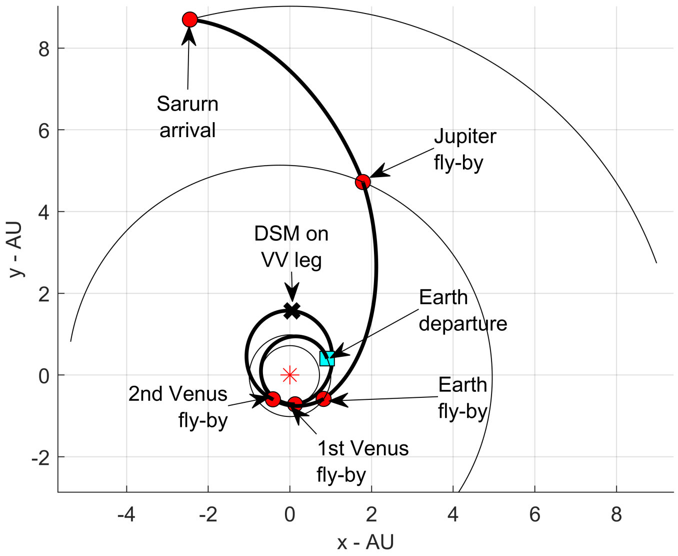


Figure 3.5: EVVEJS Cassini-like transfer within the MGA-DSM model (no DSM on the first leg).

Figure 3.5 shows the corresponding trajectory in the ecliptic plane. This mission requires one large DSM on the VV leg of about 0.4358 km/s, to increase the infinity velocity at the next Venus encounter. In all the other legs, the DSMs are in the order of 1×10^{-5} to 2×10^{-5} km/s (i.e., virtually null). The infinity velocities at the Earth departure and Saturn arrival are 4.103 km/s and 5.453 km/s, respectively.

It can be shown that having a manoeuvre in the first leg of the transfer is not essential unless the following fly-by is resonant [31,241]. In fact, having a DSM on a same-planet-to-same-planet leg allows for a variation in the infinity velocity at the next planetary encounter, which is usually important for specific mission applications (e.g., Juno mission). If this is not the case, then the DSM can be 'included' in the launch without compromising the representation of good trajectories. This holds when the first legs has less than one revolution, however there can be cases on which a large DSM might be needed to allow proper phasing with the second planet in the sequence. This is, for example, the case of Messenger, that is also found experimentally in later section 3.3.3.

3.3.3 MGA with multiple Deep Space Manoeuvres

The main limitation of the MGA-DSM model is that only one DSM is assumed on each interplanetary leg. Although many practical missions are captured within this model, missions might still require multiple impulses on a single leg, especially if multiple revolutions are involved in the transfer. For example, designing missions to Mercury similar to Messenger [132] or BepiColombo [133] might benefit from a model that takes into consideration multiple DSMs between consecutive swing-bys. This could be particularly useful in the last phases of the transfer, when multiple swing-bys with Mercury are expected in conjunction with multiple DSMs to leverage the infinity velocity at the planet, thus allowing a lighter orbiter injection. Different strategies can be found in literature to model multiple DSMs, like multiple-shooting algorithms [25] or primer-vector-based optimization [242]. These manoeuvres are useful to perform plane change, alongside changes in the argument of periapsis, or even to approximate low-thrust transfers [17]. In the following, a model that builds upon the presented MGA-DSM model is described. This model, called thereafter MGA-nDSM, assumes that a user-defined number n of DSMs can occur in each leg of the MGA trajectory. A good choice for n can be to assume a number of DSMs equal to the number of revolutions that the spacecraft performs on a planet-to-planet transfer. Thus, if the spacecraft does $N_{rev} < 2$ revolutions, then 1 DSM is included, if $2 \leq N_{rev} < 3$ then 2 DSMs are included, and so on.

The variables that define the launch event $(v_{\infty,dep}, u, v)$ and the fly-bys (r_p, ζ) are the same of the MGA-DSM model. On each leg i of the transfer, of time duration T_i, the following variables are needed to model n DSMs:

- $\eta_{i,1},...,\eta_{i,n}$ that are the fractions of time T_i at which the $1^{st}, 2^{nd}, ..., n^{th}$ DSM are performed (analogously to the MGA-DSM case).
- $\Delta\vec{v}_{i,1} = [\Delta v_x, \Delta v_y, \Delta v_z]^T_{i,1}, ..., \Delta\vec{v}_{i,n-1} = [\Delta v_x, \Delta v_y, \Delta v_z]^T_{i,n-1}$ that are the components of the $n-1$ DSMs needed on the leg. Each manoeuvre is separated by a propagation arc that brings the spacecraft from one manoeuvre to the next one. A Lambert arc is solved from the position of the last manoeuvres up to the next planetary encounter, and the n^{th} DSM is computed as velocity difference analogously to the MGA-DSM model.

The dimension of the problem is much higher than the MGA-DSM model, as this is $d = 6 + \sum_{i=1}^{n_{int}-1}(4n_i - 3) + 2(n_{int} - 2)$ where n_i are the number of manoeuvres on each leg i of the transfer. If it is assumed that no manoeuvres occur in the first leg, then $d = 2 + \sum_{i=1}^{n_{int}-1}(4n_i - 3) + 2(n_{int} - 2)$.

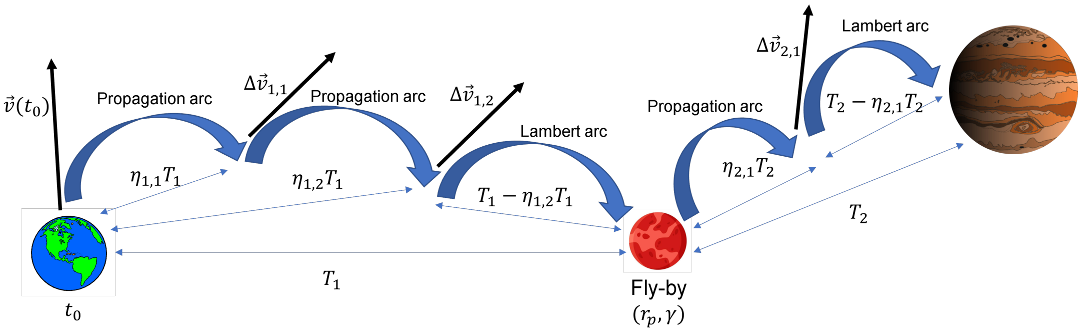


Figure 3.6: Illustration of the MGA-nDSM model.

Figure 3.6 shows a representation of an EMJ transfer with 2 DSMs on the first leg and 1 DSM on the second leg. In particular, the process to compute an MGA-nDSM trajectory is the following:

i. The step (i) is the same as the MGA-DSM model, thus it is not repeated. The initial state of the spacecraft is therefore known: $[\vec{r}(t_0), \vec{v}(t_0)]$.

ii. This state is then (analytically) propagated by integrating Eq. 3.2 from t_0 to $\eta_{1,1}T_1$, i.e., until the point of the first manoeuvre is the leg. The state is now: $[\vec{r}(t_0 + \eta_{1,1}T_1), \vec{v}^-(t_0 + \eta_{1,1}T_1)]$. If multiple manoeuvres are present in this leg, then:

a. one applies the $\Delta\vec{v}_{1,1} = [\Delta v_x, \Delta v_y, \Delta v_z]^T_{1,1}$, thus the state becomes $[\vec{r}(t_0 + \eta_{1,1}T_1), \vec{v}+ -(t_0 + \eta_1 T_1)]$, where $\vec{v}+ -(t_0 + \eta_{1,1}T_1) = \vec{v}^-(t_0 + \eta_{1,1}T_1) + \Delta\vec{v}_{1,1}$. This is propagated by an amount $\eta_{2,1}T_1 - \eta_{1,1}T_1$ until the next manoeuvre. This procedure is repeated until the last manoeuvre in the given leg.

iii. If the spacecraft has reached the point at which the last DSM is performed, the state is $[\vec{r}(t_0 + \eta_{1,n}T_1), \vec{v}^-(t_0 + \eta_{1,n}T_1)]$. Now, similarly to the MGA-DSM case, a Lambert arc is computed to find the velocity $\vec{v}^+(t_0 + \eta_{1,n}T_1)$ that brings the spacecraft from $\vec{r}(t_0 + \eta_{1,n}T_1)$ to the position of the next planet in the sequence that is $\vec{r}_{pl}(t_0 + T_1)$. The last DSM on the leg is thus computed as $\vec{v}^+(t_0 + \eta_{1,n}T_1) - \vec{v}^-(t_0 + \eta_{1,n}T_1)$. The arrival state of the spacecraft at the planet is $[\vec{r}(t_0 + T_1), \vec{v}^-(t_0 + T_1)]$, where $\vec{r}(t_0 + T_1) = \vec{r}_{pl}(t_0 + T_1)$.

iv. The fly-by is computed as in the MGA-DSM model (see step (iv)).

v. For each successive leg in the transfer, steps (ii) to (iv) are repeated until the last planet is reached.

In the EMJ example, the length of the y vector would be $d = 6 + (5 + 1) + 2 = 14$.

A mission towards Mercury similar to Messenger [132] or BepiColombo [133] can be designed. Inspired by the challenges posed by the so-called Messenger full mission proposed in the ESA GTOP database [1], an optimization of the sequence EVVYYYY (Y being Mercury) has been performed ($n_{int} = 7$). First, a grid-based optimization (see Chapter 5) is employed to quickly scan the launch window (the whole year 2005 is considered, i.e.,

[1] `https://www.esa.int/gsp/ACT/projects/gtop/messenger_full`, last accessed August 2022

the same of Messenger), trying to minimize the sum of infinity velocities at the departure and arrival plus any DSM needed in the mission. In the grid-based optimization, only 1 DSM is assumed on each leg occurring right after each fly-by. Subsequently, solutions coming from the grid-based method are used to bound a particle swarm optimization to reconstruct all the parameters needed to model the trajectory in the MGA-nDSM model. Figure 3.7.a shows the guess from the grid optimization, while Figure 3.7.b illustrates the fully optimized trajectory (black crosses in the image represent DSMs, and DSMs below 0.01 km/s are not shown). It can be seen that the fully optimized solution does not lie far from the guess, as a first proof of the goodness of the pipeline presented in Chapter 5 and expanded in successive Chapters. As a result of a dynamic programming-based optimization using defects model as from [239] and Chapter 5, the number of revolutions on each leg of the transfer are 1,3,2,2,1 and 3, thus the same number of manoeuvres are included in each leg (i.e., 1 DSM on the first leg, 3 DSMs on the second leg, and so on), for a total of 12 DSMs. Therefore, the dimension of the vector y is d=46. Details of the trajectory as from the MGA-nDSM model are reported in Table 3.1.

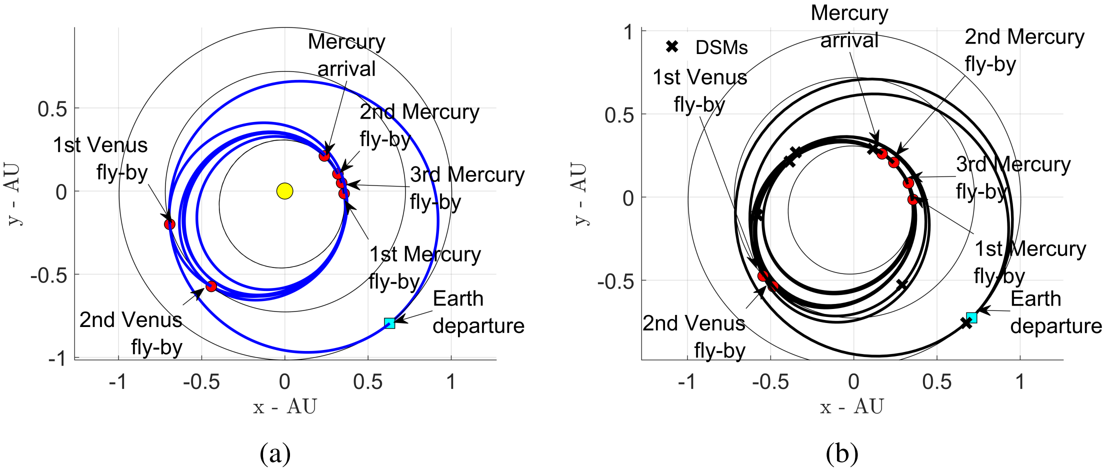


Figure 3.7: EVVYYYY Messenger-like mission as from the dynamic programming-based optimization using defects model (a) and in the MGA-nDSM model.

Table 3.1: Results for Messenger-like mission compared to solutions from defect model and refinement. When a manoeuvre is not present between two planetary encounters, a '$--$' is included.

Event	Defects solution	Refined solution
Earth departure	Jul. 31, 2005	Aug. 06, 2005
$v_{\infty,dep}$	3.77 km/s	3.00 km/s
Δv_1	$--$	1.52 km/s
Venus fly-by	Oct. 16, 2006	Oct. 31, 2006
Δv_2	1.239 km/s	0.567 km/s
Δv_3	$--$	0.00294 km/s
Δv_4	$--$	0.163 km/s
Venus fly-by	Jan. 31, 2008	Jan. 28, 2008
Δv_5	0.395 km/s	0.000649 km/s
Δv_6	$--$	0.0431 km/s
Mercury fly-by	Dec. 30, 2008	Dec. 30, 2008
Δv_7	1.094 km/s	0.152 km/s
Δv_8	$--$	0.0168 km/s
Mercury fly-by	Sep. 24, 2009	Oct. 1, 2009
Δv_9	0.193 km/s	0.219 km/s
Mercury fly-by	Jun. 13, 2010	Jun. 19, 2010
Δv_{10}	0.898 km/s	0.00136 km/s
Δv_{11}	$--$	0.000299 km/s
Δv_{12}	$--$	0.285 km/s
Mercury arrival	Jun. 6, 2011	Jun. 1, 2011
$v_{\infty,arr}$	3.81 km/s	3.73 km/s

3.4 Conclusions

This Chapter described in detail the mathematical framework and variables' set that will be employed in the rest of the thesis. These correspond to the first and third building blocks as from Table 1.2. In particular, the well-known MGA-DSM model is described that allows for computing many real-world mission scenarios.

A model that extends the MGA-DSM is also presented to allow multiple DSM on each planet-to-planet leg. This is called MGA-nDSM, and it has been shown how it can be employed to design complex mission scenarios as a Messenger-like transfer to Mercury. The main difficulty associated to such transfers is that multiple spacecraft revolutions are needed in conjunction with DSMs to leverage the infinity velocity at the target planet, to enable a lighter approach.

Chapter 4

MGA Sequences via Tisserand Graph

The first step of the solution of MGA trajectory design, i.e., the combinatorial optimization of feasible sequences, usually enters within the logic of multi-fidelity process [30], which can be either explicit or implicit. One should in fact notice that the combinatorial optimization of MGA sequences can either be made a priori, i.e., before the search of the launch date and transfer times, or within the same logic of the search space exploration (see Chapter 5), with neither conceptual nor computational difference. Here, an explicit exploration is presented for the sake of simplicity when describing the framework, although an implicit variant is also discussed in Chapter 5.

As discussed in Chapter 2, Tisserand graph [154] represents a valid option to quickly assess the feasibility of different sequences connecting two celestial objects. Most of the approaches to Tisserand graphs make use of simplified dynamical model for swing-by objects, assuming circular-coplanar orbits around the main body attractor, as well as zero-inclination orbits for the spacecraft transfer arcs. Therefore, compared to existing literature, while presenting the Tisserand graphs and their usage to generate viable MGA sequences, this Chapter also discusses their exploitation to achieve highly inclined heliocentric orbits. The work has been developed in support to the Dolphin mission [75], proposed to the 2022 ESA F/M-class call.

This Chapter articulates as follows: section 4.1 presents the mathematical steps to generate Tisserand graphs, illuminating the relationship between different model types (namely patched conics and CR3BP). This section also shows the exploitation of resonant orbits to change spacecraft energy connecting orbits that are not reachable with a single fly-by at a given planet. Section 4.2 describes the steps necessary for an automatic exploration of the Tisserand graph, also accounting for DSMs that allow to leverage the infinity velocity at planetary encounters. Section 4.3 provides practical applications of Tisserand graph exploration illustrating the capability of identifying globally optimal paths as well as reaching high-inclination orbits exploiting Venus and Earth successive fly-bys. Section 4.4 discusses main benefits and limitations of Tisserand graphs. Finally, section 4.5 derives the conclusions.

4.1 Generation of the Tisserand Graph

Tisserand graph has its roots in the Tisserand criterion [243], originally employed to identify variations on comets' orbital parameters induced by Jupiter's gravity when passing by the planet. It can be shown [244] that the following function of comets' semi-major axis (a), eccentricity (e) and inclination (i) remains constant before and after the fly-by with the planet:

$$T = \frac{r_{pl}}{a} + 2\cos(i)\sqrt{\frac{a}{r_{pl}}(1-e^2)} \tag{4.1}$$

where r_{pl} is the planet distance from the Sun. In other words, two comets observed at different epochs sharing the same value of T, known as the Tisserand invariant, are in fact the same comet observed before and after the fly-by with a planet. When designing spacecraft MGA trajectories, this process is reversed, as one is interested in identifying all the possible spacecraft orbits, in terms of (a, e, i) that can be achieved through a swing-by with a given planet. One should highlight that Eq. 4.1 is obtained within the CR3BP model [244], on which the following assumptions hold:

1. The planetary orbit is assumed circular about the centre of the main-attracting body (e.g., the Sun).
2. The masses of the Sun and planet, with mass m_1 and m_2, respectively, dominate the gravity field when compared to comet's mass.

On the one hand, although the first hypothesis might sound rather strong (see section 4.4), in most practical applications (section 4.3) the swing-by objects are characterized by relatively low orbital eccentricities, allowing (idealized) relation in Eq. 4.1 to be still very useful when employed in conjunction with higher-fidelity force models (see Chapter 5 and Chapter 6). On the other hand, also the second hypothesis is reasonable as typical masses of objects (comets or spacecraft) are several order of magnitudes lower than large bodies (planets or planetary moons) employed for swing-bys.

One notices that Eq. 4.1 is a remarkable result, describing a relationship between the CR3BP and the patched-conic model employed in this work. This relationship is illuminated by the following discussion in section 4.1.1. Eq. 4.1 is in fact obtained from the definition of the Jacobi constant of the CR3BP that is (in non-dimensional units [1]):

$$J_C = 2\left(\frac{1-r_1}{\rho_1} + \frac{r_1}{\rho_2} - 1\right) + \frac{1}{a} + 2\cos(i)\sqrt{\frac{a}{r_{pl}}(1-e^2)} \tag{4.2}$$

where r_1 is the distance of m_1 from the centre of mass of the system $M = m_1 + m_2$, and ρ_1 and ρ_2 are the distances of the satellite from m_1 and m_2, respectively. If $m_1 >> m_2$ (as in most practical applications), then $r_1 \approx 0$, and $\rho_1 \approx 1$, i.e., the satellite and planet positions coincide at the moment of fly-by, then one has that:

[1] The non-dimensional units are obtained considering: $r_1 + r_2 = 1$ and $m_1 + m_2 = 1$

$$J_C \approx \frac{1}{a} + 2\cos(i)\sqrt{\frac{a}{r_{pl}}(1-e^2)} \tag{4.3}$$

that is the non-dimensional form of Eq. 4.1.

One also highlights here that the application of Tisserand graphs is broader than the patched conics approach, and it spans from the analysis of orbital evolution of small celestial objects (e.g., asteroids), study of resonant orbits with planetary moons (e.g., at Jupiter or Saturn), or study of ballistic orbital injection at moons (e.g, Ganymede or Callisto).

4.1.1 Derivation of Tisserand Graph Equations

To assess the effect on spacecraft orbital parameters induced by a close passage with a large object, one wants to find the relationship between the parameters (v_∞, α, k) describing the encounter with a planet (see also Figure 4.1) with the spacecraft orbital elements (a, e, i). Three equations are searched for this purpose. These are obtained following the procedure adapted from [244]. This analysis assumes a crossing orbit from which the relative velocity (or v_∞) may be derived. Tisserand's equation also applies for non-crossing orbits, where crossing refers to orbits that either intersect a planet's orbit or pass through its SOI – in which case the v_∞ based formulation may not be used as v_∞ is then undefined.

Figure 4.1 illustrates the anatomy of a planetary encounter and the relationship between spacecraft and planet velocities, i.e. $\vec{v}$ and $\vec{v}_{pl}$, respectively, that is the spacecraft velocity relative to the planet $\vec{v}_\infty$, also known as infinity velocity. The angle α measures the tilt between $\vec{v}_{pl}$ and $\vec{v}_\infty$, ranging between 0 to 180 degrees. The angle k, defined from -90 to 90 degrees, measures the rotation of the plane described by the triangle of $\vec{v}$, $\vec{v}_{pl}$ and $\vec{v}_\infty$ with respect to the plane of the gravity-assist body, identified by planetary position $\vec{r}_{pl}$ and velocity $\vec{v}_{pl}$. From Figure 4.1, $k = \arccos(\hat{p}_3 \cdot \hat{c}_3)$, where $\hat{c}_3 = \vec{v}_{pl} \times \vec{v} / ||\vec{v}_{pl} \times \vec{v}||$.

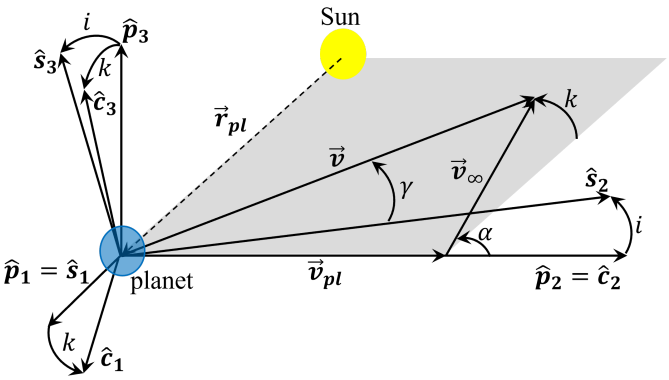


Figure 4.1: Velocity triangle for a planetary encounter and reference frames.

The reference frame $P = (\hat{p}_1, \hat{p}_2, \hat{p}_3)$ is then defined as [2]:

$$\begin{cases} \hat{p}_1 = \hat{r}_{pl} \\ \hat{p}_2 = \hat{v}_{pl} \\ \hat{p}_3 = \hat{p}_1 \times \hat{p}_2 \end{cases} \tag{4.4}$$

where $\hat{r}_{pl} = \vec{r}_{pl}/||\vec{r}_{pl}||$ and $\hat{v}_{pl} = \vec{v}_{pl}/||\vec{v}_{pl}||$ are unitary vectors in the direction of planetary position and velocity, respectively (recall that $\hat{p}_1$ and $\hat{p}_2$ are orthogonal because it is assumed circular planetary orbit). The angle k thus defines a reference frame $C = (\hat{c}_1, \hat{c}_2, \hat{c}_3)$ such that:

$$\begin{cases} \hat{c}_3 = \vec{v}_{pl} \times \vec{v}/||\vec{v}_{pl} \times \vec{v}|| = \sin(k)\hat{p}_1 + \cos(k)\hat{p}_3 \\ \hat{c}_2 = \hat{p}_2 \\ \hat{c}_1 = \hat{c}_2 \times \hat{c}_3 = \cos(k)\hat{p}_1 - \sin(k)\hat{p}_3 \end{cases} \tag{4.5}$$

on which the vector is conveniently written as:

$$\vec{v}_\infty = -v_\infty \sin(\alpha)\hat{c}_1 + v_\infty \cos(\alpha)\hat{c}_2 \tag{4.6}$$

[2] Quantities without vector sign represent the magnitude, i.e. $||\vec{(\cdot)}|| = (\cdot)$

A clockwise rotation of $P = (\hat{p}_1, \hat{p}_2, \hat{p}_3)$ about $\hat{p}_1$ of an angle i, i.e., the spacecraft orbital inclination, allows to identify the spacecraft orbital plane that defines the frame $S = (\hat{s}_1, \hat{s}_2, \hat{s}_3)$ such that:

$$\begin{cases} \hat{s}_1 = \hat{p}_1 \\ \hat{s}_2 = \cos(i)\hat{p}_2 + \sin(i)\hat{p}_3 \\ \hat{s}_3 = -\sin(i)\hat{p}_2 + \cos(i)\hat{p}_3 \end{cases} \tag{4.7}$$

In such frame, the spacecraft velocity vector $\vec{v}$ is conveniently written as:

$$\vec{v} = v\sin(\gamma)\hat{s}_1 + v\cos(\gamma)\hat{s}_2 \tag{4.8}$$

where γ is the spacecraft flight-path angle measured between $\hat{s}_2$ and $\vec{v}$. Combining Eq. 4.5 and 4.6, one has:

$$\vec{v}_\infty = -v_\infty \sin(\alpha)\cos(k)\hat{p}_1 + v_\infty \cos(\alpha)\hat{p}_2 + v_\infty \sin(\alpha)\sin(k)\hat{p}_3 \tag{4.9}$$

And using Eq. 4.7 and 4.8:

$$\vec{v} = v\sin(\gamma)\hat{p}_1 + v\cos(\gamma)\cos(i)\hat{p}_2 + v\cos(\gamma)\sin(i)\hat{p}_3 \tag{4.10}$$

Moreover, since $\vec{v}_\infty = \vec{v} - \vec{v}_{pl}$ and $\vec{v}_{pl} = v_{pl}\hat{p}_2$ as in Eq. 4.4, substituting Eq. 4.10, one finds that:

$$\vec{v}_\infty = v\sin(\gamma)\hat{p}_1 + (v\cos(\gamma)\cos(i) - v_{pl})\hat{p}_2 + v\cos(\gamma)\sin(i)\hat{p}_3 \tag{4.11}$$

Therefore, combining Eq. 4.9 and 4.11:

$$\begin{cases} -v_\infty \sin(\alpha)\cos(k) = v\sin(\gamma) \\ v_\infty \cos(\alpha) = v\cos(\gamma)\cos(i) - v_{pl} \\ v_\infty \sin(\alpha)\sin(k) = v\cos(\gamma)\sin(i) \end{cases} \tag{4.12}$$

Rewriting the second equation as $v_\infty(\cos(\alpha)+\frac{v_{pl}}{v_\infty})$, and dividing by the third one, one finds:

$$\sin(k) = \tan(i)\left(\frac{\cos(\alpha)+\frac{v_{pl}}{v_\infty}}{\sin(\alpha)}\right) \tag{4.13}$$

Which is the first relation needed to assess the effect of a gravity assist with the spacecraft orbit at the planetary encounter, as it links fly-by parameters (v_∞, α, k) with the spacecraft inclination i.

The second relation can be obtained by considering the cosine-law between $\vec{v}$, $\vec{v}_{pl}$ and $\vec{v}_\infty$:

$$v^2 = v_\infty^2 + v_{pl}^2 + 2v_\infty v_{pl}\cos(\alpha) \tag{4.14}$$

Through the vis-viva equation, it is possible to relate v with the spacecraft semi-major axis a, $v^2 = \frac{2\mu}{r} - \frac{\mu}{a}$, where μ is the gravitational parameter of the Sun and r is the Sun-spacecraft distance which coincides with r_{pl} in patched-conic approximation. Under the assumption of circular planetary obits around the Sun, one uses the circular velocity $v_{pl} = \sqrt{\mu/r_{pl}}$ to find the following relation:

$$\left(\frac{v}{v_{pl}}\right)^2 = \left(2-\frac{r_{pl}}{a}\right) \tag{4.15}$$

Then, combining Eq. 4.14 and 4.15, one has:

$$\frac{r_{pl}}{a} = 2 - \frac{1}{v_{pl}^2}(v_\infty^2 + v_{pl}^2 + 2v_\infty v_{pl}\cos(\alpha)) \tag{4.16}$$

that is the second relation needed to assess the effect of a gravity assist on spacecraft orbital elements (a,e,i). In fact, Eq. 4.16 relates the semi-major axis a with parameters (v_∞,α).

The third relation can be obtained by squaring Eq. 4.11 and by recalling that $\sqrt{\mu a(1-e^2)} = rv\cos(\gamma)$ through the definition of the specific angular momentum of spacecraft orbit $\vec{h} = \vec{r}\times\vec{v}$. In this way, one has:

$$\left(\frac{v_\infty}{v_{pl}}\right)^2 = \left(\frac{v}{v_{pl}}\right)^2 + 1 - 2\cos(i)\sqrt{\frac{a}{r_{pl}}(1-e^2)} \tag{4.17}$$

Where $r = r_{pl}$ and $v_{pl} = \sqrt{\mu/r_{pl}}$ have been suitably employed. Combining Eq. 4.15 and 4.17, one finally has:

$$3 - \left(\frac{v_\infty}{v_{pl}}\right)^2 = \frac{r_{pl}}{a} + 2\cos(i)\sqrt{\frac{a}{r_{pl}}(1-e^2)} \tag{4.18}$$

which is the third relation needed. Right-hand side of Eq. 4.18 coincides with the Tisserand invariant T defined in Eq. 4.1, thus one has that $T = 3 - \left(\frac{v_\infty}{v_{pl}}\right)^2$, or in terms of the Jacobi constant from Eq. 4.3 $J_C \approx 3 - \left(\frac{v_\infty}{v_{pl}}\right)^2$, illustrating the relationship between CR3BP and patched-conic model and showing that the infinity velocity magnitude remains constant during a close passage with a planetary swing-by.

To sum up, Eq. 4.13, 4.16 and 4.18 are the searched relations and are here reported:

$$\begin{cases} \sin(k) = \tan(i)\left(\frac{\cos(\alpha)+\frac{v_{pl}}{v_\infty}}{\sin(\alpha)}\right) \\ \frac{r_{pl}}{a} = 2 - \frac{1}{v_{pl}^2}(v_\infty^2 + v_{pl}^2 + 2v_\infty v_{pl}\cos(\alpha)) \\ 3 - \left(\frac{v_\infty}{v_{pl}}\right)^2 = \frac{r_{pl}}{a} + 2\cos(i)\sqrt{\frac{a}{r_{pl}}(1-e^2)} \end{cases} \tag{4.19}$$

These permit to visualize spacecraft orbit in terms of (a,e,i) parametrized with respect to (v_∞, α, k). For a given v_∞, any fly-by with a planet moves the spacecraft orbit along the constant v_∞ rotating the vector $\vec{v}_\infty$ of an amount δ and $\Delta k = k^+ - k^-$ [3]. One recalls from Chapter 3 that the angle δ represents the rotation of vector $\vec{v}_\infty$ around $\vec{v}_{pl}$ and it is governed by v_∞ and the hyperbola periapsis during the fly-by (see again Chapter 2). Figure 4.2 shows the spherical relationship between these three angles, governed by vectors $\vec{v}_{pl}$, $\vec{v}_\infty^-$ and $\vec{v}_\infty^+$.

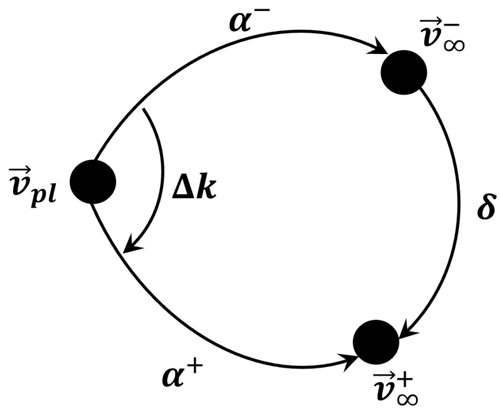


Figure 4.2: Spherical relationship between $\vec{v}_{pl}$, $\vec{v}_\infty^-$ and $\vec{v}_\infty^+$ (vectors pointing out of the plane).

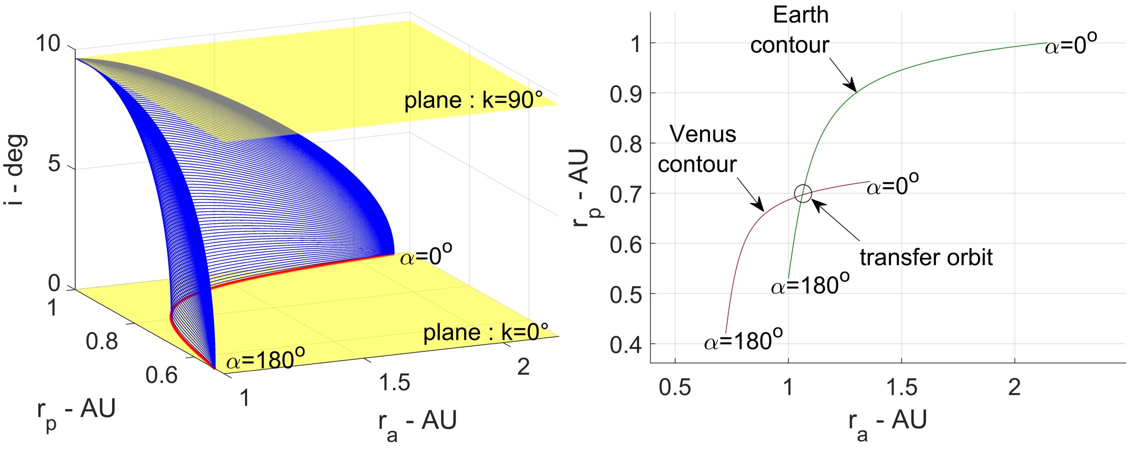
Spherical cosine-law solved for Δk yields to the following relationship:

$$\cos(\Delta k) = \frac{\cos(\delta) - \cos(\alpha^+)\cos(\alpha^-)}{\sin(\alpha^+)\sin(\alpha^-)} \tag{4.20}$$

which highlights the dependency of Δk from α^+ and α^-. For different values of (v_∞, α, k), from Eq. 4.19 one then obtains different spacecraft orbital loci, that can be represented in a map as in Figure 4.3. Specifically, Figure 4.3a shows the relationship between spacecraft apoapsis $r_a = a(1+e)$, periapsis $r_p = a(1-e)$ and inclination i for different encounter options at the Earth in terms of (α, k) for fixed infinity velocity $v_\infty = 5$ km/s. On the one hand, if $\alpha = 0$ deg, then the spacecraft velocity is aligned with the planet one, corresponding to the highest orbital energy for a given v_∞ magnitude; if $\alpha = 180$ deg, the spacecraft

[3] Superscripts '+' and '-' refer to quantities before and after the fly-by

velocity is anti-parallel to the planet's one, resulting in the lowest orbital energy for the given v_∞. On the other hand, a planetary encounter with $k = 0$ deg corresponds to an $i = 0$ deg, while the maximum inclination is achieved at $k = 90$ deg in correspondence of $\alpha = 90$ deg.



(a) (b)

Figure 4.3: Relationship between apoapsis, periapsis and inclination for a fixed $v_\infty = 5$ km/s at the Earth (a). Plot (b) represents infinity velocity contours of 5 km/s at Venus and Earth on the plane $k = 0$ deg.

This is a very important consideration, as one notices from the first relation in Eq. 4.19 that the maximum achievable inclination with a planetary swing-by is limited by the spacecraft relative velocity to the planet, i.e., v_∞, no matter how many swing-bys the spacecraft performs with the given planet. Therefore, if the aim of the mission is to maximize the spacecraft orbital inclination, according to Eq. 4.19 one should maximize the v_∞ at the planetary encounter and perform a fly-by such that $k^+ = k^- + \Delta k = 90$ deg. Later section 4.3.2 discusses such mission options in detail. From Figure 4.3.b, intersections between contours associated to different planetary encounters correspond to transfer orbits between the two planets, i.e., to orbits that cross two planets at the same time. However, this opportunity only exists from an energetic point of view, since Eq. 4.19 contains no explicit information regarding planetary phase and transfer times. This is in fact the main limitation of Tisserand graphs, as it is discussed in later section 4.4. An MGA sequence on the Tisserand graph would thus look like as a sequence of intersections between contours (see also section 4.2), i.e., as a sequence of orbits that connects two planetary orbits.

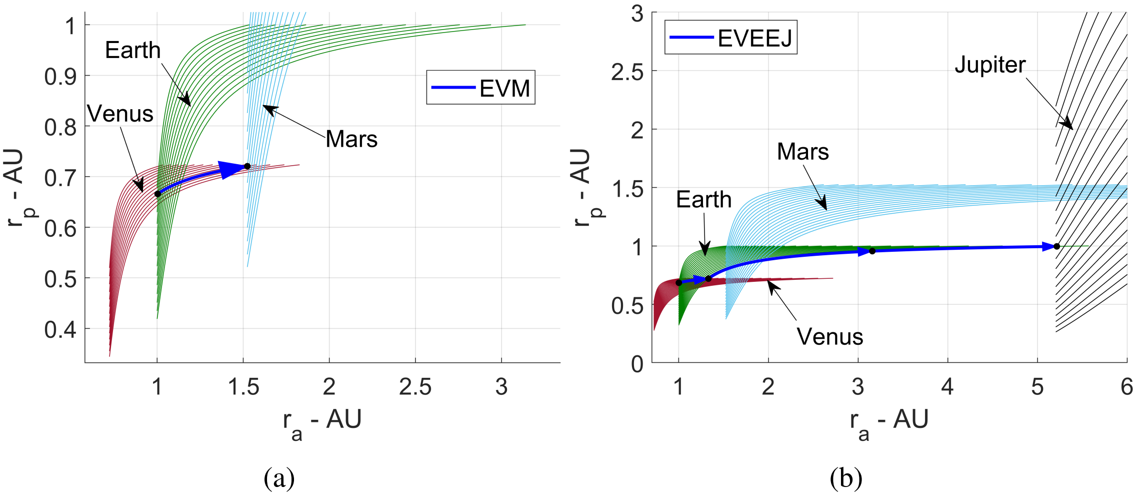


Figure 4.4: Earth-Venus-Mars (a) and Earth-Venus-Earth-Earth-Jupiter (b) paths on the Tisserand graph. The v_∞ have discrete values of $3, 3.3, 3.6...$ km/s (increasing downwards).

Moving along the Tisserand graph thus consists in linking different intersections through planetary fly-bys. This implies a variation in the angle α along a given planetary contour that is induced by the fly-by with that planet. Figure 4.4a shows an Earth-Venus-Mars (EVM) path that starts at the intersection between Earth and Venus contours at infinity velocity equal to 3.3 km/s and 6 km/s, respectively, moves along Venus contour at 6 km/s, i.e., performing the fly-by with Venus, and ends at the next intersection with Mars at 6 km/s. The maximum variation of α along any contour is limited by the minimum fly-by altitude admissible for the given planet as in Eq. 3.3. Thus, two consecutive intersections can by linked with a single fly-by only if $\delta = \alpha^+ - \alpha^-$ does not exceed its maximum value, occurring at the minimum swing-by altitude for a given v_∞ as in Eq. 3.3. If this condition is not satisfied, multiple swing-bys with the same planet can be used to progressively modify α allowing distant intersections to be linked. This is shown in Figure 4.4b, where the sequence EVEEJ exploits a mid-course fly-by with the Earth at $v_\infty = 9$ km/s that connects the intersection between the EV contours to the one at EJ.

4.1.2 Resonant Loci on Tisserand Graph

Since the change of the orbit induced by a single fly-by is limited by the minimum altitude the spacecraft can have during the manoeuvre, a single fly-by may be insufficient to reach the next planet. Table 4.1 reports minimum altitudes that are typically employed for Solar System planets [154, 174].

Table 4.1: Minimum gravity-assist altitudes and periapsis for Solar System planets (until Saturn).

Planet	h_{min} **(km)**
Mercury	200
Venus	200
Earth	200
Mars	200
Jupiter	349555
Saturn	116464

One thus needs to find intermediate orbits that are reachable on the contour, as with resonant transfers. These transfers allow the spacecraft to perform a 360 degrees revolution (or multiples) around the Sun on a same planet-to-planet transfer, hence re-encountering the planet for a consecutive gravity assist. On resonant transfers, a ratio of integers exists between the planet and the spacecraft orbit periods. The ratio is expressed as N:M, where N and M are the number of planet and spacecraft revolutions, respectively.

For Solar System planets, the most common resonances are listed in Table 4.2, alongside with practical examples of actual space missions making use of specific resonant ratios. As an example, missions towards Jupiter, like Galileo [170], can usefully exploit the 2:1 resonance on consecutive Earth fly-bys to increase the apoapsis on an overall EVEEJ strategy. Resonances could be also used in conjunction with DSMs to increase or decrease the infinity velocity magnitude with respect to a given planet [187]. This could be useful to leverage the planet-spacecraft relative velocity (see also section 4.2.2), to allow for a more convenient spacecraft capture around a given planet or even planetary moon, such as Europa [167–169] or Enceladus [176–178].

Table 4.2: Common resonances for innermost Solar System planets.

Planet	Resonance	Examples
Mercury	0.5:0.5	BepiColombo phasing with Mercury
	1:1	BepiColombo and Messenger energy reduction [25]
	6:5	BepiColombo energy reduction and v_∞ leveraging [25]
	5:4	BepiColombo energy reduction and v_∞ leveraging [25]
	4:3	BepiColombo energy reduction and v_∞ leveraging [25]
	3:2	BepiColombo and Messenger energy reduction [25]
Venus	1:1	BepiColombo energy reduction and Solar Orbiter energy modification [25, 173]
	2:1	Cassini like transfer for energy increase and v_∞ leveraging manoeuvre with DSM [171]
	3:4	BepiColombo energy reduction and Solar Orbiter plane changing and energy reduction [25, 173]
	2:3	Solar Orbiter plane changing [25, 173]
	1:2	Solar Orbiter plane changing [25, 173]
Earth	1:1	JUICE plane change [182] and v_∞ leveraging
	2:1	Galileo energy increase [170] and v_∞ leveraging
	3:1	Uranus Pathfinder energy increase [245] and v_∞ leveraging
	2:3	Solar Orbiter option for energy modification [173]
	3:2	Potential use for plane change manoeuvre
Mars	1:1	Potential use for asteroid/comet rendezvous
	2:1	Potential use for asteroid/comet rendezvous
	3:1	Potential use for asteroid/comet rendezvous and energy increase for low-v_∞ Jupiter approaches

The N:M ratio defines a strong relation between the planet and spacecraft revolutions around the Sun:

$$T_{res} = \frac{N}{M} T_{pl} \tag{4.21}$$

where T_{res} is the orbital period of the spacecraft resonant orbit and T_{pl} is the planetary period around the Sun. It is thus possible to compute the spacecraft semi-major axis a_{res} by inverting $T_{res} = 2\pi\sqrt{a_{res}^3/\mu}$. For any given v_∞ one can then use the second equation of 4.19 to find the corresponding α. The searched relation is thus:

$$\alpha_{res} = \arccos\left(\frac{v_{pl}}{2v_\infty}\left(1-\left(\frac{v_\infty}{v_{pl}}\right)^2-\frac{r_{pl}}{a_{res}}\right)\right) \tag{4.22}$$

Not all the resonant orbits defined by the N:M ratio are always possible. In fact, from Eq. 4.22 a resonant orbit is achievable for a given v_∞ only if the argument of the arc-cosine is between -1 and 1, i.e., only if:

$$\left|\frac{v_{pl}}{2v_\infty}\left(1-\left(\frac{v_\infty}{v_{pl}}\right)^2-\frac{r_{pl}}{a_{res}}\right)\right| \leq 1 \tag{4.23}$$

One can finally use Eq. 4.19 to find all the relevant orbital parameters for a given resonant ratio, e.g., (r_a, r_p, i) , by parametrizing with respect to v_∞ and k. Figure 4.5 represents a Tisserand graph (k=0) with defined resonant orbits at Venus, Earth, and Mars.

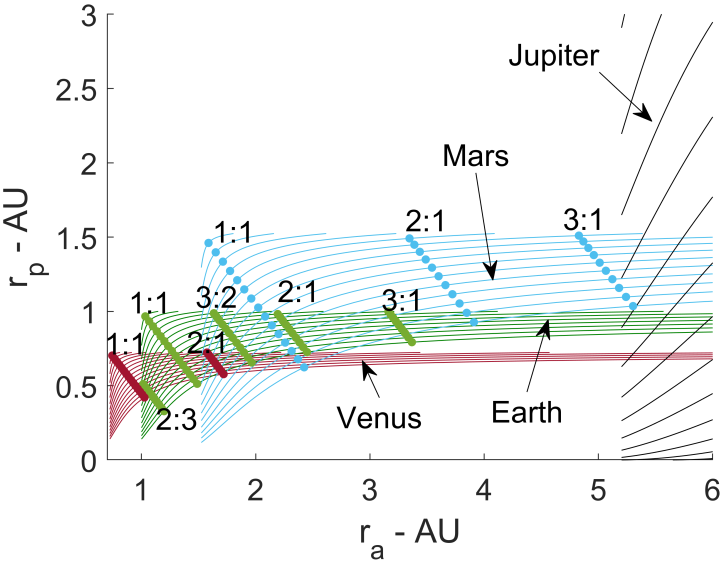
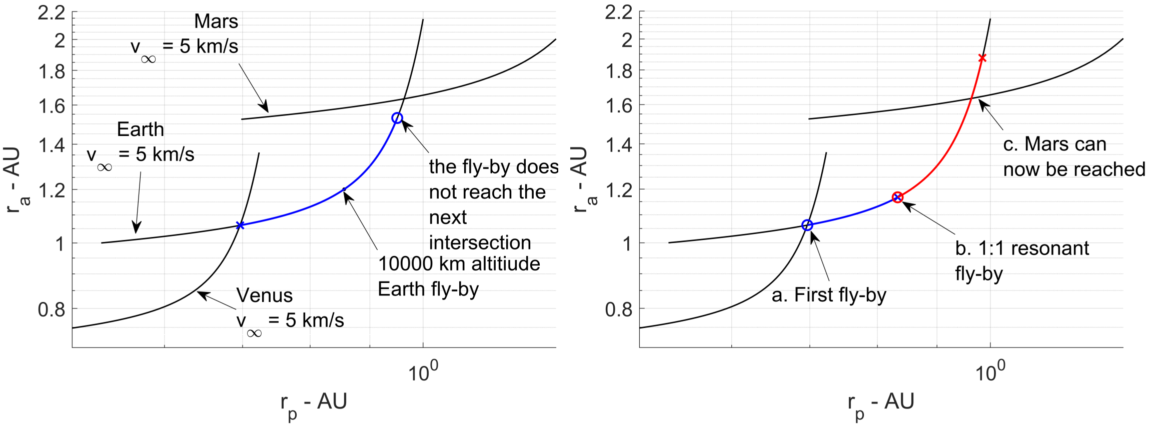


Figure 4.5: Orbits at different v_∞ at Venus, Earth, Mars, and Jupiter. Some resonant orbits are also represented for Venus, Earth, and Mars. The v_∞ have discrete values of 3,4,5. . . km/s (increasing downwards).

Figure 4.6 shows an example of the exploitation of resonant transfers to connect Venus contour at 5 km/s and Mars contour at 5 km/s. Assuming a minimum fly-by altitude of 10000 km at the Earth (v_∞=5 km/s), a single fly-by is not sufficient to reach the next intersection. The 1:1 resonance is thus used so that Mars can now be reached.



(a)

(b)

Figure 4.6: Exploitation of 1:1 resonance at Earth (v_∞=5 km/s) to connect Venus (v_∞=5 km/s) and Mars contours (v_∞=5 km/s). [181]

4.2 Exploration of the Tisserand Graph

This section deals with the exploration of the Tisserand graph to construct planetary sequences. This is the core of the AUTOMATE toolbox [181], as defined in Chapter 1. Exploring a Tisserand graph consist in evaluating the effect of all the possible sequences of planetary swing-bys in the parameters of the Tisserand invariant. It is thus possible to identify all the planetary sequences and excess velocities which are energetically feasible to reach the desired target orbit by connecting contours of two different planets when intersections occur. Three steps are necessary and are tackled in the following: (1) the generation of all the intersections between different planetary contours, (2) the infinity leveraging manoeuvres to change the v_∞ with respect to a planet, and (3) the generation of MGA sequences linking successive intersections along a given contour.

4.2.1 Constructing the Intersections

As stated, the aim is to connect the intersections between infinity velocity contours of different planets by means of successive 'jumps' induced by the planetary gravity along each line that modify the α angle (according to the minimum fly-by altitude constraint). Intersections between contours on the Tisserand graph are determined by solving the non-linear problem:

$$r_{p1}(E) - r_{p2}(E) = 0 \tag{4.24}$$

where r_{p1} and r_{p2} are the periapsis radii on the two contours, expressed as functions of the orbital energy E. These are obtained through $r_p = a(1-e)$ from Eq. 4.19 for any v_∞ and v_{pl} (provided $k = 0$ deg $\Rightarrow i = 0$ deg). One recalls that the orbital energy E is related to the semi-major axis a through $E = -\mu/(2a)$. Eq. 4.24 has been explicitly written as function of the orbital energy E to stress the fact that the intersections on Tisserand map, i.e., transfers orbits between two planets, only exist from an energetic point of view, with no explicit consideration of transfer time or planetary phasing.

Algorithm 1 Pseudo-code for computing all the intersections between any infinity velocity contour of different planets.

```
Select the planets for fly-bys (ordered by their distance from the Sun [4]), and a set of infinity velocities:
for each planet $P_i$ do
    for each infinity velocity $v_\infty$ in the set do
        Generate the $v_\infty$ contour for $P_i$ (Eq. 4.19)
        for each planet $P_j$ such that $j > i$ do
            for each infinity velocity $v_\infty$ in the set do
                Generate the $v_\infty$ contour for $P_j$ (Eq. 4.19)
                Solve Eq. 4.24 with Newton-Raphson method to find intersections
                Compute and store the orbits (Eq. 4.19)
            end for
        end for
    end for
end for
```

It is now possible to compute all the intersections between different contours for any planet and for any v_∞, as shown in Algorithm 1. As it can be seen, one needs to define a set of infinity velocities at which the spacecraft encounters any planet. In theory, any set would contain infinite values for v_∞ (since it is a continuous-varying variable), however a

[4] For example, if V, E and M are selected, one has an ordered vector: $[P_1, P_2, P_3] = [V, E, M]$.

common practice is to discretise the values that v_∞ can assume between a minimum and a maximum. The choice of such extremal values, alongside the step size, is rather arbitrary and depends upon the application. A typical way of proceeding would be to assign extremal values that are ample enough to account for a variety of encounters (say between 1 to 20 km/s) and step sizes of at least one order of magnitude less than the corresponding planet orbital velocity (e.g., 0.5-1 km/s for innermost planets like Mercury, Venus, and Earth, and 0.5 km/s for slower planets like Jupiter, Saturn, Uranus and Neptune). Such values are enough to represent a wide variety of planetary sequences, also finding references solutions and globally optimal paths (see section 4.3). Figure 4.7 highlights all the possible intersections between Earth and Mars computed by means of Algorithm 1.

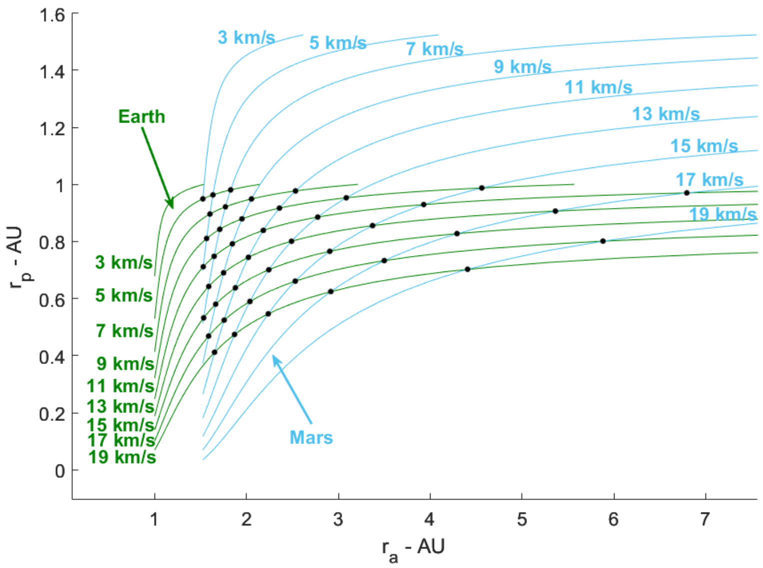


Figure 4.7: Intersections between Earth and Mars contours. The v_∞ have discrete values explicitly reported in the plot.

4.2.2 Changing the Relative Velocity at a Planetary Encounter

A fly-by with a given planet does not change the infinity velocity at the encounter, which remains constant (in magnitude) as demonstrated by Eq. 4.19. However, some missions might still benefit of infinity velocity changes with respect to a given planet mainly to:

- Reduce the relative speed to lower the magnitude of the orbit insertion manoeuvre at the target planet (e.g., Messenger [132], BepiColombo [133], Europa orbiter studies [167–169])

- Increase the relative speed to escape innermost regions of the system (e.g., Juno [246], Cassini [171])

In order to change the relative velocity with a given planet a DSM is needed, which allow to perform a so-called v-infinity leveraging transfer (VILT) [178, 187]. It is assumed that: (1) the DSM is performed at one of the two apses of the transfer orbit, and that it is aligned with the orbital velocity, (2) the VILTs are applied on resonant orbits in between consecutive fly-bys with the same object. The anatomy of the VILT is represented in Figure 4.8 [5].

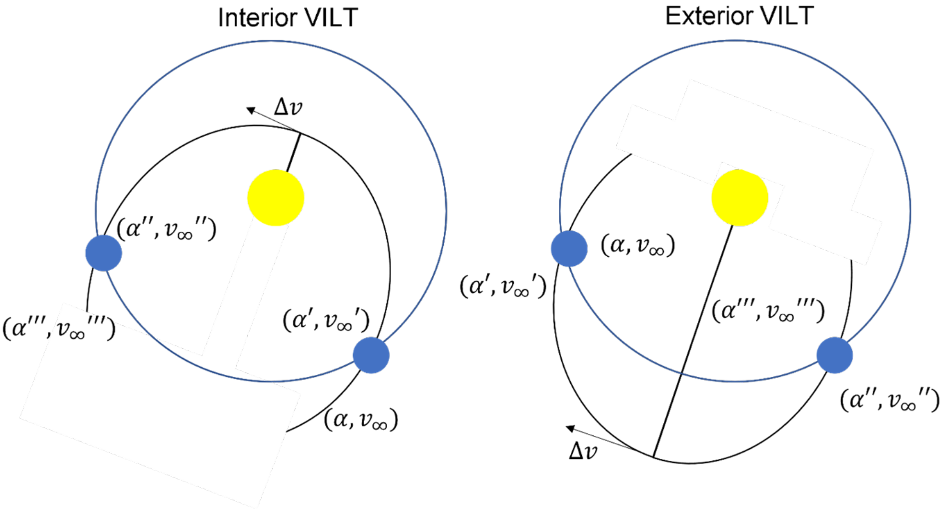


Figure 4.8: Anatomy of interior and exterior VILTs. Note that multiple revolutions can happen.

The problem is connecting two intersections on the Tisserand map, namely (α, v_∞) and (α''', v_∞'''), such that $v_\infty \neq v_\infty'''$. The idea is that the spacecraft encounters a planet at given conditions (α, v_∞), it executes a fly-by and reaches the orbit (α', v_∞'), with $v_\infty = v_\infty'$. The spacecraft then performs a DSM either at the apoapsis (exterior VILT) or at the periapsis (interior VILT) that allows to jump from (α', v_∞') to (α'', v_∞''). A fly-by is then performed

[5] Figure 4.8 is for representation only: multiple revolutions can occur on VILT manoeuvres.

such that (α''',v_∞''') is achieved, implying $v_\infty'' = v_\infty'''$, that is only if $|\alpha''' - \alpha''| \leq \delta_{max}$, where δ_{max} is a function of v_∞''.

The parameters defining the DSM are thus $v_\infty = v_\infty'$, v_∞'' (fixed parameters), α' and k_{ei} (free parameters), the latter being $k_{ei} = -1$ or $k_{ei} = +1$ if the DSM is interior or exterior, respectively. The parameter α', i.e., the post-fly-by angle, is free as the VILT is considered phase-free, i.e., the planet is not assured to be re-encountered after the manoeuvre. If one wants to include the phasing constraint, and thus to fix the post-fly-by orbit, the procedure presented in [178] should be followed. This assumes that the spacecraft is injected into a quasi-resonant orbit N:M after the first fly-by and waits a number of revolutions $L < M$ before performing the manoeuvre, such that the next encounter with the planet is enforced. In any case, the α' must still obey to the maximum-deflection rule, i.e., $|\alpha' - \alpha| \leq \delta_{max} = f(v_\infty)$. An example of VILT manoeuvre is shown in Figure 4.9, on which a Δv manoeuvre (applied at the apoapsis in this case) changes the periapsis around the 2:1 resonant locus with Venus, to jump from 7 km/s to 9 km/s with respect to the planet. This allows to connect the Earth contours at 4 km/s and 9 km/s.

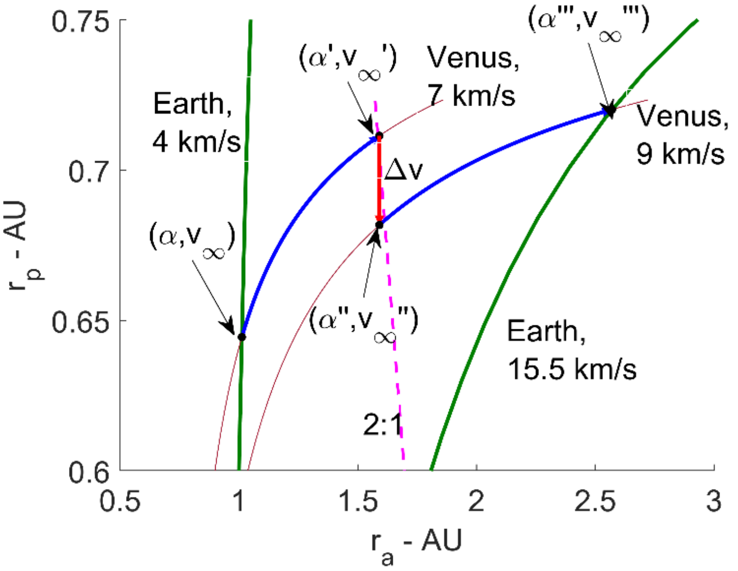


Figure 4.9: Example of a DSM to change the infinity velocity at Venus from 7 km/s to 9 km/s. A quasi-resonant 2:1 orbit is exploited for this purpose.

The point at which the spacecraft performs the manoeuvre is the leveraging apse, and its distance from the central body is:

$$r_{la} = a(1 + k_{ei}e) \tag{4.25}$$

Solving for e and substituting into the third equation of 4.19, one gets the following quadratic equation:

$$\frac{4ar_{la}^2}{r_{pl}^3} - \frac{8a^2r_{la}}{r_{pl}^3} + \frac{a^2T^2}{r_{pl}^2} - \frac{2aT}{r_{pl}} = 0 \tag{4.26}$$

where $T = 3 - (v_\infty/v_{pl})^2$ is the Tisserand parameter. Therefore, one starts with a given (α, v_∞) in the Tisserand map, e.g., at an intersection between two contours, which implies $v_\infty = v'_\infty$. The leveraging apse r_{la} can be computed from Eq. 4.25 as a function of a' (or α' using Eq. 4.19), for a given k_{ei}. For the post-manoeuvre orbit, r_{la} remains constant, thus one can use Eq. 4.26 to solve for a'', i.e., the semi-major axis of the post-manoeuvre orbit, for a given $T'' = 3 - (v''_\infty/v_{pl})^2$.

The resulting Δv is given by the velocity difference of the two orbits (before and after the manoeuvre, i.e., (α', v'_∞) and (α'', v''_∞), respectively) at the leveraging apse:

$$\Delta v = \left| \sqrt{\frac{2\mu}{r_{la}} - \frac{\mu}{a'}} - \sqrt{\frac{2\mu}{r_{la}} - \frac{\mu}{a''}} \right| \tag{4.27}$$

As stated, the Δv would be a function of a' (or α' using Eq. 4.19) since no planetary phasing is enforced for the transfer. Thus, two options are available:

- Either one selects the minimum Δv provided that is below a tolerance Δv_{tol} (e.g., 0.5 km/s)
- Or one solves the phasing problem for a' (or α') [178]

Algorithm 2 presents the procedure for computing the VILT manoeuvres that are useful to link two intersections on the Tisserand map that do not share the same infinity velocity with a given planet.

Algorithm 2 Pseudo-code for computing VILT transfers.

```
Select the starting point $(\alpha,v_\infty)$, arrival point $(\alpha''',v'''_\infty)$, a set of target infinity velocities (after the DSM) and $\Delta v_{tol}$
for each $v''_\infty$ in the set do
    for $\alpha' \in [0,180]$ deg do
        if $|\alpha' - \alpha| \leq \delta_{max}(v_\infty)$ then
            Find $(a',e')$ from $(\alpha',v'_\infty = v_\infty)$ (Eq. 4.19)
            for $k_{ei} = +1,-1$ do
                Compute $r_{la}$ (Eq. 4.25)
                Find $a''$ (Eq. 4.26)
                Compute minimum $\Delta v$ (Eq. 4.27)
                if $\Delta v \leq \Delta v_{tol}$ then
                    Compute $\alpha''$ from $(a'',e'')$ (Eq. 4.25 and 4.19)
                    if $|\alpha''' - \alpha''| \leq \delta_{max}(v''_\infty)$ then
                        Save $(\alpha',v'_\infty)$, $(\alpha'',v''_\infty)$ and $\Delta v$
                    end if
                end if
            end for
        end if
    end for
end for
```

A typical example for the use of VILTs is the exploration of moons' systems around the gas giants, namely Jupiter and Saturn. Figure 4.10 shows an example of a tour in the Jovian system obtained on Tisserand graph. The aim is to rendezvous with the moon Europa at the lowest v_∞ possible, as this impacts the Δv for inserting a spacecraft around the moon. The tour departs at a high eccentric orbit at Ganymede at approximately 7 km/s of relative velocity with the moon [25]. Figure 4.10.a shows the approach at Europa using successive resonant fly-bys at Ganymede in 8:1, 4:1 and 5:2 resonant ratios. Then fly-bys at Europa and Ganymede allows for gradual reduction of the infinity velocity at Europa up until 3 km/s. The leveraging tour is then computed using Algorithm 2, such that successive pseudo-resonant transfers with Europa at 5:3, 3:2, 5:4 and 11:10 are built in conjunction with DSMs that allow the infinity velocity jumps. This is shown in Figure 4.10.b. This phase allows to reach 0.5 km/s of relative velocity at Europa, with a total Δv of 452 m/s.

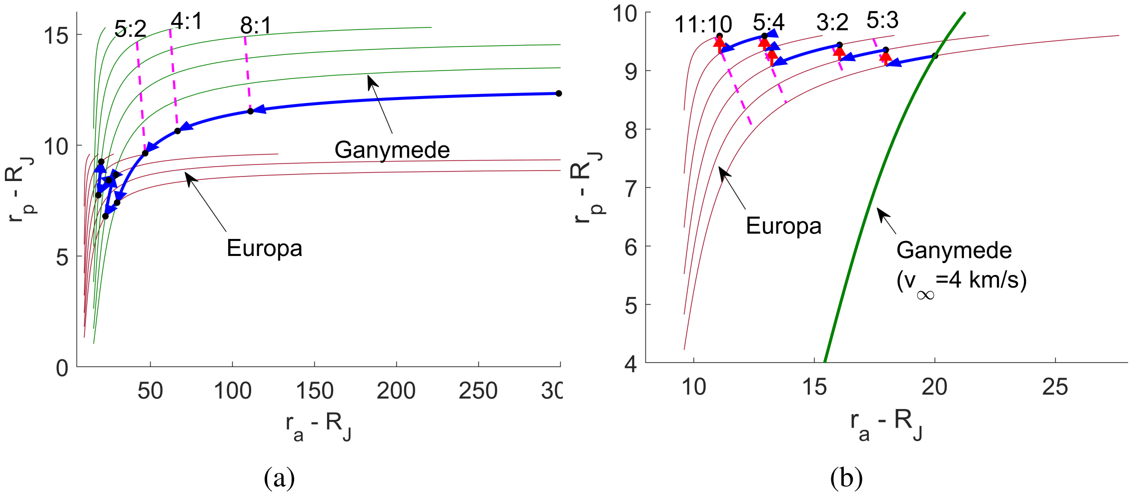


Figure 4.10: Figure (a) shows a possible tour with Ganymede and Europa fly-bys to reduce the infinity velocity at Europa up until 3 km/s. The infinity velocity contours go from 1 km/s to 7 km/ with step size of 1 km/s. Figure (b) shows the VILT tour at Europa. Red arrows represent DSMs. The infinity velocity contours go from 0.5 km/s to 3 km/ with step size of 0.5 km/s. Axes are normalized with respect to Jupiter radius R_J.

In Appendix D, a more complex scenario with 49 fly-bys with Saturn moons is also presented. To prevent factorial explosion of trajectory options in terms of sequences of fly-bys and DSMs, dynamic programming (see also Chapter 5) is used to find optimal solutions (either single- or multi-objective) while mitigating computer memory issue.

4.2.3 Finding MGA Sequences

The last step of the exploration of a Tisserand graph consists in connecting intersections along a given planetary infinity velocity contour, i.e., by performing either a fly-by with the given planet or a DSM exploiting existing resonances. As it can be seen from Figure 4.7, a given infinity velocity contour associated to a planet might intersect with multiple contours of another planet. This suggests that a tree representation of the possible intersections might be beneficial in order to conveniently explore the graph. Let's consider a planetary sequence that is ABC. On a tree-like Tisserand graph, each node encodes the minimum set of variables that define an intersection, which are $[(\alpha, v_\infty)_A, (\alpha, v_\infty)_B]$, namely α and v_∞ for the two different planets A and B that are being connected. Each edge of the tree-like Tisserand graph connects two intersections by means of a fly-by, namely $[(\alpha, v_\infty)_A, (\alpha^-, v_\infty)_B] \rightarrow [(\alpha^+, v_\infty)_B, (\alpha, v_\infty)_C]$.

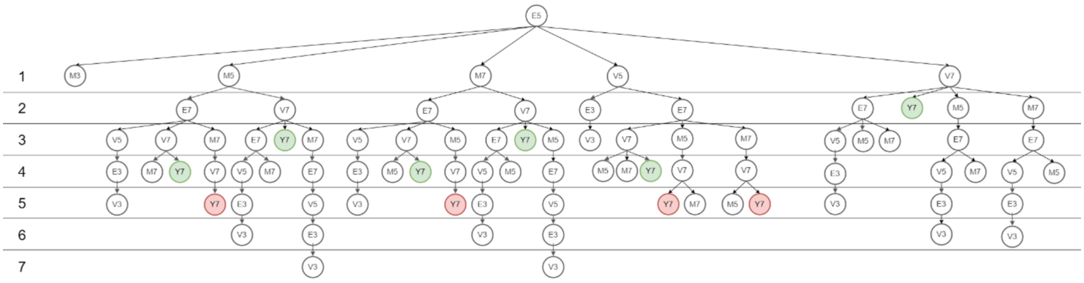

E5
1 M3 M5 M7 V5 V7
2 E7 V7 E7 V7 E3 E7 E7 Y7 M5 M7
3 V5 V7 M7 E7 Y7 M7 V5 V7 M5 E7 Y7 M5 V3 V7 M5 M7 V5 M5 M7 E7 E7
4 E3 M7 Y7 V7 V5 M7 E7 E3 M5 Y7 V7 V5 M5 E7 M5 M7 Y7 V7 V7 E3 V5 M7 V5 M5
5 V3 Y7 E3 V5 V3 Y7 E3 V5 Y7 M7 M5 Y7 V3 E3 E3
6 V3 E3 V3 E3 V3 V3
7 V3 V3

Figure 4.11: Tree structure for Tisserand graph. [181]

Figure 4.11 shows the tree-like structure of Tisserand graph. In this case, the planets considered are Mercury (Y), Venus (V), Earth (E), and Mars (M). The infinity levels grid, for the sake of simplicity, is only made by 3, 5 and 7 km/s contours at each planet. One wants to reach Mercury after an Earth departure set at v_∞ =5 km/s, namely E5 in the image. All the contours that intersect E5 are those with Mars at 3, 5 and 7 km/s (i.e., M3, M5 and M7 in Figure 4.11) and Venus at 5 and 7 km/s (i.e., V5 and V7). This set of intersections (namely M3, M5, M7, V5, V7) corresponds to the first depth-level of the tree. Starting from all the intersections found, another depth-level can be constructed by finding all the intersections with the ones at the previous level. One proceeds by adding levels until a maximum depth is reached. If the maximum depth is set to 4, all the paths finishing with the nodes represented in green will be saved in the output. The paths finishing in with a red node are not, as the depth of these nodes is greater than 4 (more layers are shown in Figure 4.11 for representation purposes). In this case, six paths are found with four different sequences: EMEVY, EMVY, EVEVY and EVY. In this case, one does not optimize mission analysis metrics, but simply lists all the feasible MGA sequences that reach Mercury with the given set up.

The number of combinations to be explored is usually manageable by a complete enumeration strategy [162] for interplanetary flights, since a maximum of 4-5 swing-bys is sufficient to reach any object in the Solar System, and it is also an ample enough range resulting in transfer times in the order of 5 to more than 20 years (see also Chapter 5). A breadth-first exploration [162] is used here. One can envisage mission options for which incomplete approaches might be needed (see [167–169] or Appendix D).

Algorithm 3 Pseudo-code for planetary sequences generation on Tisserand map.

```
Select the departing planet, departing conditions, planets for fly-bys, set of infinity
velocities, maximum tree depth and $\Delta v_{tol}$
Compute all the resonances using Eq. 4.21-4.23
Compute all the intersections between planets (Algorithm 1)
Compute all the resonant orbits (see Table 4.2 and Eq. 4.21-4.22)
for all the intersections that satisfy the departing conditions do
    while the maximum tree depth is not reached do
        Extract intersections reachable from current ones along the given $v_\infty$ contours
        for those that cannot be reached with a single fly-by do
            if resonances are available then
                Perform fly-bys exploiting the resonant transfer
                Save the transfers
            end if
            Perform as many fly-bys as needed to achieve the next intersection
        end for
        for intersections that do not share the same infinity velocity contour do
            Connect the intersections by means of DSMs (Algorithm 2)
        end for
    end while
end for
```

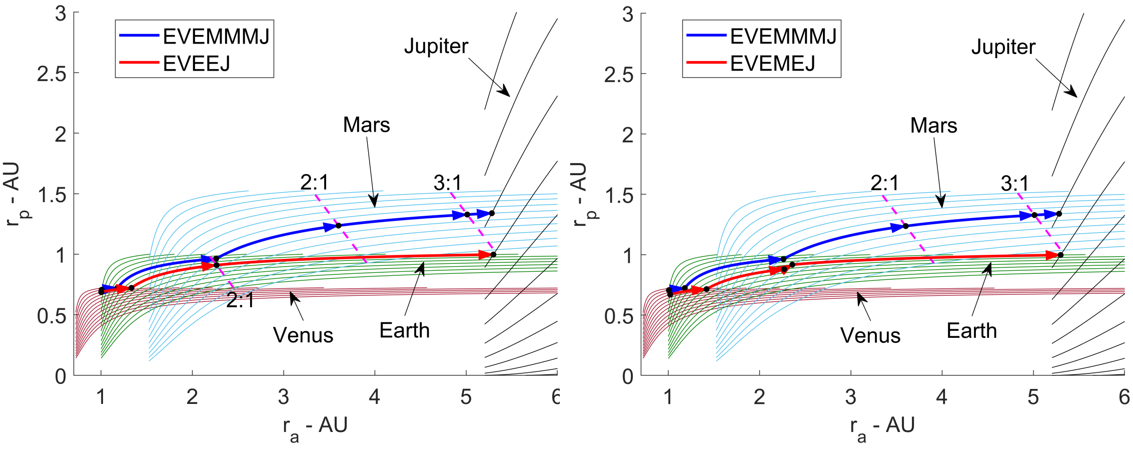
Algorithm 3 summarizes the main steps of the generation of sequences from Tisserand graphs. It starts by defining the departing planet, with desired departing conditions (e.g., $v_\infty \leq 5$ km/s), the desired arrival planet, the maximum number of legs of the overall MGA sequences, and the v_∞-levels set-up (minimum and maximum value, and step size). Algorithm 1 is to compute all the intersections for the given set up. The tree is then built, only saving those intersections that can be reached within the maximum deflection allowed for the given fly-by condition. If this condition is not met, intermediate fly-bys between two intersections are included (both exploiting resonances and performing as many maximum-deflection fly-bys as needed). If a resonance is to be exploited, the algorithm checks if DSM can be performed to change the infinity velocity at the planet, and to reach another intersection (Algorithm 2).

4.3 Practical Applications

In this section, several mission scenarios are discussed by means of the processes described in this Chapter. In particular, sequences towards Jupiter are analysed in section 4.3.1 to assess the efficiency of proposed algorithms to replicate referenced transfers, as well as to identify sequences that well correspond to the theoretical Δv-minimum solutions. Sections 4.3.2 and 4.3.3 show how Tisserand graphs can be helpful to design complex MGA sequences that allow access to high-inclination orbits.

4.3.1 Earth-Jupiter Missions

Tisserand graphs are very convenient obtain a wide range of planetary encounters which are feasible from an energetic point of view, and one finds transfers that approximate global optimality for a given transfer scenario. An exemplar application is the Earth (E) – Jupiter (J) scenario, where an automatic exploration of the Tisserand map finds 27 sequences with 3 to 5 fly-bys for an infinity velocity range of 3 to 5 km/s at the Earth and up to 6 km/s at Jupiter with step sizes of 1 km/s between different contour lines. Few seconds are needed to scan the Tisserand map and list the MGA sequences on standard laptop (i.e., 4 GHz single core).



(a) (b)

Figure 4.12: EVEMMMJ sequence compared to EVEEJ (a) and EVEMEJ (b).

Despite the relatively high number of trajectory options to be analyzed in successive optimization steps, the convenient search space transcription in conjunction with dynamic programming approach (see Chapter 5) leads to manageable computational effort (see again Chapter 5) so that a robust and efficient exploration of the whole search space can be performed neither with a priori knowledge on the solution nor the need of stochastic meta-heuristic strategies.

Among all the trajectory options, Figure 4.12 shows that sequences like EVEMMMJ well approximate the theoretical global optimum from Hohmann-like transfers for a mission from Earth to Jupiter, when compared to well-known transfers such as EVEEJ, as in Figure 4.12.a, and EVEMEJ, as in Figure 4.12.b, i.e., sequences considered, for example, for Galileo mission [170] or JUICE [6]. In fact, having successive Mars fly-bys, in resonance with the planet on a 2:1 and 3:1 resonant ratio on each consecutive MM legs, allows to approximate the theoretical Hohmann transfer model, with a near-half ellipse transfer from Earth to Venus on the first leg and a near-half ellipse from Mars to Jupiter on the last leg. Such global optimum solution can be analyzed intuitively from the fact that a typical cost function that considers the sum of all the manoeuvres in the transfer (similar to the later Eq. 4.29, see also Chapter 5) [6] is dominated by departing and arrival infinity velocities ($v_{\infty,dep}$ and $v_{\infty,arr}$, respectively). In each transfer, the lowest $v_{\infty,dep}$ to reach Venus ranges between 2.9 and 3.5 km/s depending on launch date. For the EVEEJ and EVEMEJ cases, the lowest $v_{\infty,arr}$ at Jupiter estimated from a Hohmann transfer from Earth to Jupiter is approximately 5.6 km/s. However, for the EVEMMMJ case, the Mars gravity assist raise the perihelion of the spacecraft trajectory so that the final leg lies close to a Hohmann transfer between Mars and Jupiter, where the lowest possible $v_{\infty,arr}$ is now 4.3 km/s, which minimizes the Δv required for the mission, at the price of an extended transfer time of about 17.7 years. The trajectory in such case is shown in Figure 4.13 where the 2:1 and 3:1 resonances intervene on the successive Mars-Mars transfers.

[6] accounting for infinity velocities at departing and arrival planet

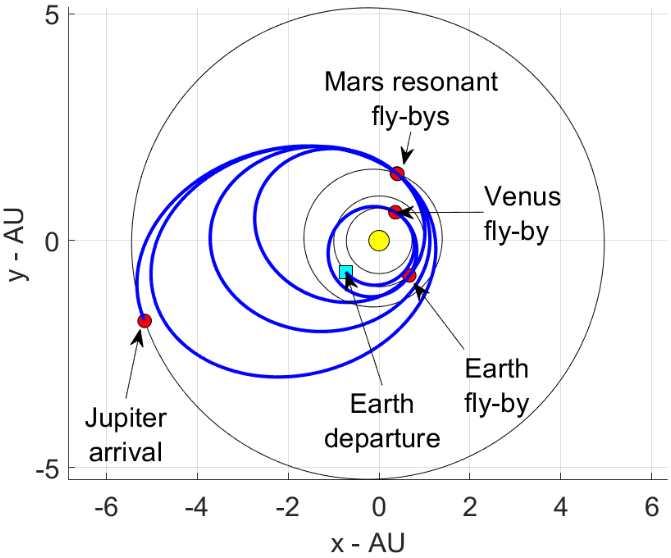


Figure 4.13: EVEMMMJ transfer exploiting 2:1 and 3:1 resonant transfers on the successive MM legs.

4.3.2 High-Inclination Orbits via Resonant Loci

Some missions might benefit from high-inclined orbits (e.g., $\geq$ 10 deg) to achieve specific objectives. This is the case of ESA's Solar Orbiter mission [173], or JUICE mission [6], or even Dolphin mission proposal [75]. Tisserand graphs can then be used to analyze different planetary sequences that allow for plane-change manoeuvres by means of successive swing-bys.

The maximum inclination i_{max} achievable after a planetary fly-by can be deduced from the first relation of Eq. 4.19:

$$i_{max} = \arctan\left(\frac{v_\infty \sin\alpha}{v_{pl} + v_\infty \cos\alpha}\right) \tag{4.28}$$

In this way, it is possible to map the maximum inclination achievable with a fly-by for different Solar System planets for any given relative velocity v_∞, as in Figure 4.14. As it can be seen from Figure 4.14, the maximum reachable inclinations increase with the distance of the objects from the Sun (Mercury being the closest and Saturn the farthest among the ones considered here), as this depends inversely upon the planetary velocity as from Eq. 4.28 (i.e., for a given relative velocity, the farther the planet, the lower its velocity with respect to the Sun, the higher the maximum inclination achievable). Figure 4.14 also shows that, in order to maximize the inclination after a planetary swing-by, the planet-spacecraft relative velocity should be the maximum possible.

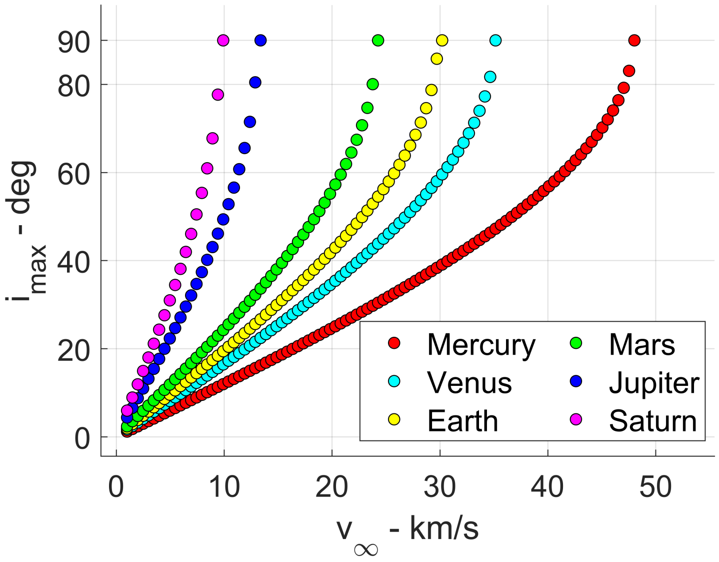


Figure 4.14: Maximum inclination achievable for different Solar System planets with respect to the spacecraft relative velocity.

It should be noted that the maximum deflection rule expressed as in Eq. 3.3 imposes a limit on the maximum achievable plane change (in terms of Δk) for a given planetary encounter. Moreover, provided a planetary encounter condition in terms of (v_∞, α), in order to maximize the Δk (i.e., to maximize the variation of inclination Δi as from the first relation in Eq. 4.1), one should maximize the deflection δ, that is inversely proportional to v_∞. Therefore, it is likely that the maximum inclination would not be achievable with a single planetary fly-by. In addition, from Eq. 4.20, the gravity of the planet influences the Δi, as the lower the gravity field, the lower the deflection δ, thus the lower the inclination change Δi. In this sense, planets like Mercury or Mars are not particularly useful to maximize inclination change Δi, due to their relatively weak gravity field (i.e., approximately one order of magnitude lower than Venus and Earth, and three with respect to Jupiter and Saturn).

4.3.3 Possible Strategies Exploiting Resonant Orbits

A possible strategy to achieve high-inclination orbits would be to increase spacecraft relative velocity with respect to a given planet (e.g., Solar Orbiter used Venus [173], NASA's Ulysses mission used Jupiter [247]), and then to perform as many fly-bys as possible with the same planet. This phase would likely be performed by means of successive resonant transfers with the planet. However, reaching relatively high values of infinity velocity (e.g., 15 to 20 km/s from Figure 4.14), is only possible by means of MGA sequences, mainly due to propellant limitations. As stated, strategies with Mercury and Mars, while potentially providing good inclinations, are not considered here mainly because of two reasons:

- The relative velocity required to reach such orbits is high, thus MGA sequences are likely to require high propellant consumption due to the rare synchronicity between Venus/Earth and the given planets.
- The gravity field of Mercury and Mars is at least one order of magnitude weaker than the other planets. This means that multiple fly-bys with those planets are required, increasing substantially the overall mission duration.

Since the aim of the paragraph is to study the accessibility of high-inclination orbits by means of resonant transfers, strategies with outer planets (Jupiter, Saturn, etc.) are not considered here due to the long mission duration implications.

4.3.3.1 Strategies with Venus: Solar Orbiter Mission

One should notice that strategies with Venus are particularly effective because of the relatively high gravity of the planet, especially if relatively extended transfer times are available. For example, Solar Orbiter [173], launched in February 2020, is expected to reach approximately 33 deg of inclination after 9 years of flight exploiting successive resonant fly-bys with Venus, trying to maximize the inclination with respect to the Solar equator, while gradually reducing its perihelion to observe the regions close to the Sun.

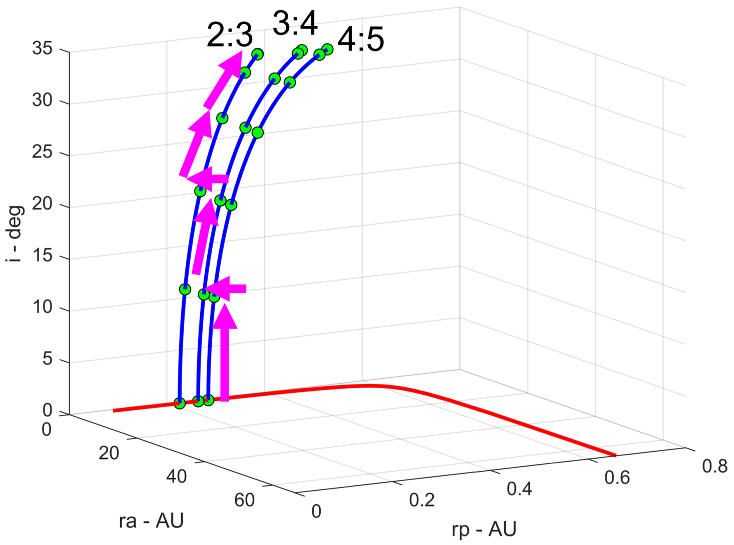


Figure 4.15: Resonant loci for an encounter with Venus at v_∞=20 km/s. Red curve is the $(v_\infty, \alpha, k = 0)$ contour. Green points separate maximum deflection fly-bys on the $k \neq 0$ plane.

The strategy adopted aims at maximizing the infinity velocity at Venus and then to perform successive resonant fly-bys with the planet to gradually increase the inclination. The required infinity velocity at Venus is approximately 20 km/s as from Figure 4.14. A quick analysis of the Tisserand graph as from section 4.2, allows to identify EVEV or its variant EVVEV as possible strategies to reach the desired v_∞ at Venus. After the last Venus encounter, the resonant tour can begin. Since one of the aim of Solar orbiter is to reduce the perihelion, resonant ratios that are lower than 1 are necessary. A possible strategy is represented in Figure 4.15. After first reaching Venus with the relative velocity considered here, the lower periapsis solution (i.e., 2:3) may not be reachable after the first fly-by, so the sequence identified by the pink arrows uses a step along the 4:5 resonance locus that allows an increase to be made in inclination. At the next fly-by, a resonance change is targeted at 3:4, whilst still increasing the inclination. A further increase in inclination can be achieved by maintaining the 3:4 resonance, however, this will increase the periapsis. Thus, at the next fly-by the 2:3 resonance is targeted twice to achieve a further inclination increase up to the required 33 deg while maintaining the periapsis at about 0.3 AU.

4.3.3.2 Strategies with Earth: Dolphin Mission

As from Figure 4.15, exploiting the Earth for planning plane-change resonant tours allows for higher inclinations with lower infinity velocity required compared to the Venus case. Moreover, the Earth has stronger gravity field than Venus, plus closer fly-by distances are allowed due to the increasing accuracy of the navigation, thus the plane change manoeuvre is likely to be more effective. However, the periapsis reachable with Earth successive resonant orbits is substantially higher than those reachable with Venus, which is why Venus still is the optimal choice for missions like Solar Orbiter. This can be seen from Figure 4.16, where the 2:3 contour allows for a minimum periapsis higher than 0.4 AU.

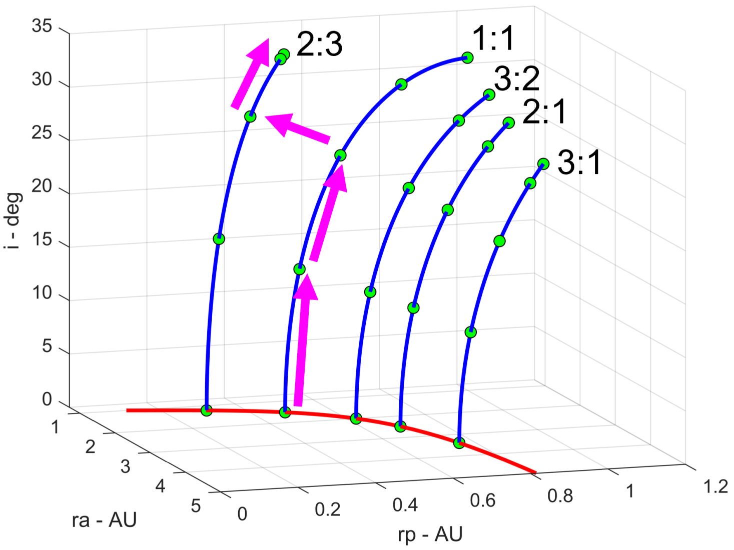


Figure 4.16: Resonant loci for an encounter with Earth at v_∞=15 km/s. Red curve is the $(v_\infty, \alpha, k = 0)$ contour. Green points separate maximum deflection fly-bys on the $k \neq 0$ plane.

In any case, the Earth still represents a valid option to increase the orbital inclination if the periapsis is not a constraint. This is the case of Dolphin mission proposal [75], on which the heliocentric inclination should be maximized to study the differences between interstellar and interplanetary dust particles. The strategy is similar to the one considered in section 4.3.3.1, on which the aim is to arrive at the Earth with the required infinity velocity (14 km/s for the example considered here). Tisserand graph exploration as from section 4.2 allows to identify EVE or EVEVE as possible strategies to reach the required velocity at the Earth, although for the EVE option the departing infinity velocity might be relatively high, i.e., in the order of 5 km/s.

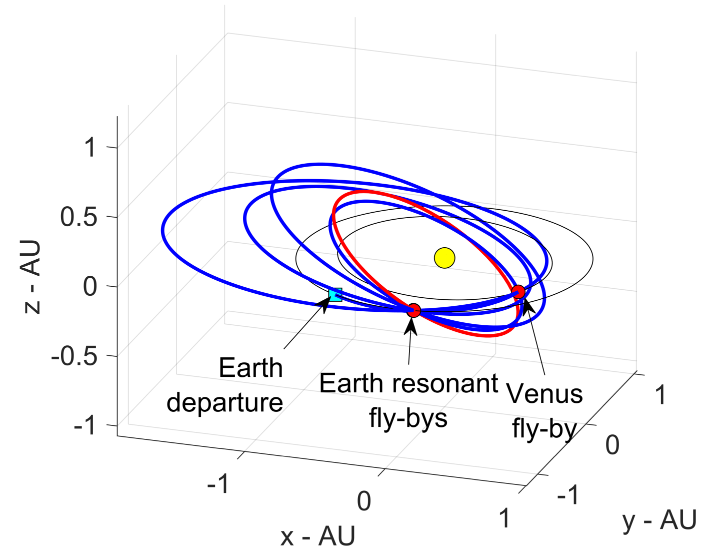


Figure 4.17: EVEEEE sequence with 1:1, 1:1, and 2:3 resonant orbits on successive EE legs. The red orbit is achieved after the last fly-by with the Earth.

Successive 1:1 resonant transfers allow to gradually increase the inclination up to about 24 degrees. Although successive 1:1 resonant orbits are available, the asymptote of the resonant locus at 1:1 is approaching, thus one recognizes that higher inclinations are achievable if the 2:3 locus is exploited. Finally, performing a last fly-by with the Earth at 15 km/s should allow for a maximum inclination of approximately 33 degrees (see Figure 4.14). Therefore, a final fly-by of the Earth is performed to maximize the final inclination. This strategy allows to reach approximately 33.34 degrees (similar value of the Solar Orbiter scientific orbit). Figure 4.17 shows the trajectory that the spacecraft would follow for the specified Dolphin-like mission option (departure in May 2031) with successive Earth resonant fly-bys, on which, the 1:1, 1:1 and 2:3 resonant transfers intervene. Such strategy allows for a relatively fast mission (5 years from launch to the last Earth encounter).

4.4 Benefits and Limitations of Tisserand Graphs

As already stated, the main benefit of Tisserand graphs is the quick analysis of possible planetary sequences to reach specific orbital regions at a very first level of fidelity, i.e., by exploiting only energetic considerations. The key part of any multi-fidelity process is thus to select the information to be passed to the next level of fidelity (discussed in Chapter 5). This section deals with this problem, particularly focusing on: (a) the infinity velocity information derivable from the Tisserand graphs (section 4.4.1) and (b) the phasing problem to estimate transfer times between two successive planets in the MGA sequence (section 4.4.2). It is shown in these sections that the information on the infinity velocities at planetary encounters is useful in successive stages to reduce the size of the search space, while the information on the phasing is not accurate due to the circular-coplanar approximation.

4.4.1 Infinity Velocity Information from Tisserand Graph

The infinity velocity information derived from the Tisserand graph is particularly useful to constraint the search for trajectories. Missions towards Jupiter and Saturn are taken as example. The setup for the Tisserand exploration problem is reported in Table 4.3.

Table 4.3: Setup for Tisserand graph exploration for missions towards Jupiter and Saturn.

Planets considered	Venus, Earth, Mars, Jupiter, Saturn
Departing planet	Earth
Arrival planet	Jupiter or Saturn
Maximum number of planets	6
Minimum v_∞ range	3 km/s
Maximum v_∞ range	17 km/s
Step size	1 km/s
Departing v_∞ range	$[3,5]$ km/s
Arrival v_∞ range	$[0,9]$ km/s

Algorithm 3 is then used to construct sequences on the map. Among the sequences, one identifies well-known trajectories like EVEMEJ and EVVEJS, similar to JUICE [6] and Cassini [171], respectively. The values for the maximum infinity velocities at each planetary encounter are reported in Table 4.4. A multi-objective dynamic programming (MODP) optimization as from [239] and Chapter 5 is then used to validate the results obtained. This considers Δv defects manoeuvres at each planetary fly-by as described in Chapter 3 as a result of incoming and outgoing Lambert arcs at any planetary encounter. The functions to be minimized are:

$$\begin{cases} f_1 &= v_{\infty,dep} + \sum_{i=1}^{n_{int}-1} \Delta v_i + v_{\infty,arr} \\ f_2 &= \sum_{i=1}^{n_{int}-1} T_i \end{cases} \tag{4.29}$$

where n_{int} is the number of planets in the sequence. The functions f_1 and represent overall f_2 consumption and transfer duration, respectively. The optimization scenario for Jupiter missions is reported in Table 4.5. The optimization scenario used for optimizing EVVEJS is reported in Table 4.6 (note that is is similar to Table 5.2 of Chapter 5, but with the additional constrain on the infinity velocities at the encounters).

Table 4.4: Maximum values for infinity velocities required to reach Jupiter and Saturn with EVEMEJ and EVVEJS.

Planet:	Earth	Venus	Earth	Mars	Earth	Jupiter
v_∞:	5 km/s	10 km/s	14 km/s	17 km/s	17 km/s	9 km/s
Planet:	Earth	Venus	Venus	Earth	Jupiter	Saturn
v_∞:	5 km/s	10 km/s	10 km/s	16 km/s	14 km/s	9 km/s

Table 4.5: Optimization scenario for EVEMEJ sequence for a launch in 2023.

Design variables	**Values and bounds**
Sequence	Known: EVEMEJ
Departure velocity magnitude	$v_{\infty,dep} \in [3,5]$ km/s
Maximum defect at each fly-by	$\Delta v \in [0,2]$ km/s
v_∞ constraints at fly-bys	See Table 4.4
Launch window	$t_0 \in [8400.5, 8765.75]$ MJD2000
Number of revolutions about the Sun	$N_{rev} = 0$
Transfer times between planets	$T_{1,2} \in [50,750]$ days $T_{3,4} \in [50,850]$ days $T_5 \in [500,2500]$ days

Table 4.6: Optimization scenario for EVVEJS sequence for a launch in 1997.

Design variables	**Values and bounds**
Sequence	Known: EVVEJS
Departure velocity magnitude	$v_{\infty,dep} \in [3,5]$ km/s
Maximum defect at each fly-by	$\Delta v \in [0,2]$ km/s
v_∞ constraints at fly-bys	See Table 4.4
Launch window	$t_0 \in [-1095.5, -730.25]$ MJD2000
Number of revolutions about the Sun	$N_{rev} = 0$
Transfer times between planets	$T_1 \in [30,400]$ days $T_2 \in [100,470]$ days $T_3 \in [30,400]$ days $T_4 \in [400,2000]$ days $T_5 \in [1000,6000]$ days

Table 4.7 summarizes the computational effort in terms of number of Lambert arcs (N_l), number of defects (N_d) to be computed, and number of routes to be stored for the MODP approach ($N_{r,MODP}$) for the cases considered (note that the information for EVVEJS is found also in Table 5.7 of Chapter 5). Including the constraint on infinity velocities as from Table 4.4 reduces considerably the number of Lambert arcs, defects, and solutions to be stored without losing information on the globally optimum path or the Pareto front information, providing a much more efficient exploration of the search space (see also later Chapter 5).

Table 4.7: Performances for EVEMEJ and EVVEJS without and with the information on infinity velocities at planetary encounters as from Tisserand graph exploration. No defects are computed on the first leg of the transfer, thus a '$--$' is included.

EVEMEJ – without Tisserand information on infinity velocities			
	N_L	N_d	$N_{r,MODP}$
EV	28548	$--$	1192
VE	14040	278298	10365
EM	41919	4516520	29247
ME	71556	862143	13450
EJ	37408	693384	25583
Totals	**193471**	**6350345**	**25583**
	Number of points in Pareto front		**190**
	Optimum for f		**8.836 km/s**
EVEMEJ – with Tisserand information on infinity velocities			
	N_L	N_d	$N_{r,MODP}$
EV	28548	$--$	1169
VE	13338	273546	9375
EM	23763	345765	28014
ME	53133	774834	11789

Continued on next page

Table 4.7 – *Continued from previous page*

EJ	26270	55104	25518
Totals	**145052**	**144924945**	**25518**
Number of points in Pareto front			**190**
Optimum for f_1			**8.836 km/s**
EVVEJS – without Tisserand information on infinity velocities			
	N_L	N_d	$N_{r,MODP}$
EV	15128	−−	995
VV	6572	123380	3403
VE	5208	79732	20302
EJ	28035	562035	8479
JS	292734	650520	1754661
Totals	**347677**	**1415667**	**1754661**
Number of points in Pareto front			**333**
Optimum for f_1			**9.494 km/s km/s**
EVVEJS – with Tisserand information on infinity velocities			
	N_L	N_d	$N_{r,MODP}$
EV	15128	−−	995
VV	6572	123380	3400
VE	5208	79360	20264
EJ	28035	558831	8260
JS	291066	648018	1747463
Totals	**346009**	**1409589**	**1747463**
Number of points in Pareto front			**325**
Optimum for f_1			**9.494 km/s**

In fact, the optimal transfers and Pareto fronts well corresponds to the actual JUICE and Cassini missions, as confirmed by Table 4.8 and Chapter 5), respectively, as an additional proof of the goodness of the whole pipeline of Chapter 4 and 5. Moreover, there is no difference in the EVEMEJ Pareto fronts, as also shown in Figure 4.18.a, while the degradation for the EVVEJS case is only appreciable for high f_1 values, as from Figure 4.18.b, that most likely do not correspond to feasible mission scenarios.

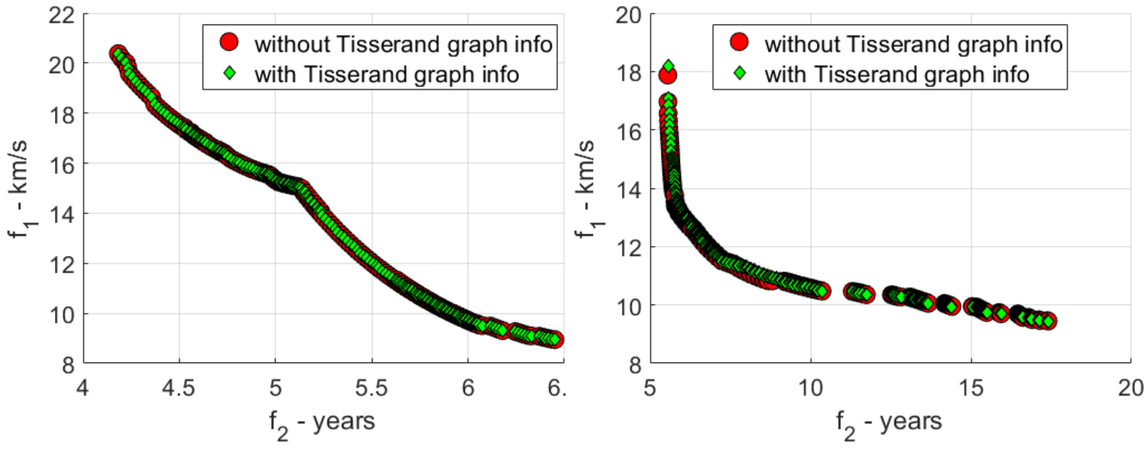
Table 4.8: Results for JUICE solution in the given mission scenario compared to solutions from defect model.

Event	**JUICE**	**Defects solution**
Earth departure	May 27, 2023	May 29, 2023
$v_{\infty,dep}$	3.18 km/s	3.18 km/s
Δv_1	−−	−−
Venus fly-by	Oct. 20, 2023	Oct. 22, 2023

Continued on next page

Table 4.8 – *Continued from previous page*

Δv_2	$--$	$--$
Earth fly-by	Aug. 30, 2024	Aug. 30, 2024
Δv_3	$--$	0.0115 km/s
Mars fly-by	Feb. 10, 2025	Feb. 12, 2025
Δv_4	$--$	$--$
Earth fly-by	Nov. 23, 2026	Nov. 23, 2026
Δv_5	$--$	$--$
Jupiter arrival	Nov. 06, 2029	Jan. 14, 2030
$v_{\infty,arr}$	5.54 km/s	5.62 km/s
f_1	8.72 km/s	8.81 km/s
f_2	6.44 years	6.63 years



(a) (b)

Figure 4.18: Comparison of EVEMEJ (a) and EVVEJS (b) multi-objective optimization without and with Tisserand graph information on infinity velocities at the encounters.

4.4.2 Phasing Problem

The use of circular co-planar approximation implied by Tisserand graphs results in differences in velocity and position with respect to the nominal planetary ephemerides [240]. Such differences make difficult to estimate the transfer time of an MGA sequence directly in the Tisserand map exploration.

One could consider the case of the EVEMEJ mission, similar to the one selected for JUICE [6]. One possibility to account for the planetary phasing is to solve the Kepler's problem for the orbit that connects Earth and Venus on the first leg [154]. However, at the next leg, i.e., Venus-Earth, one should consider the position of the previously visited planet (Earth) updated accordingly to the circular co-planar model. However, this might result in errors on the Earth position needed for next fly-by to happen, which are in the order of 20 degrees in longitude [181]. To overcome such error, a large manoeuvre on the Venus-Earth leg should be included.

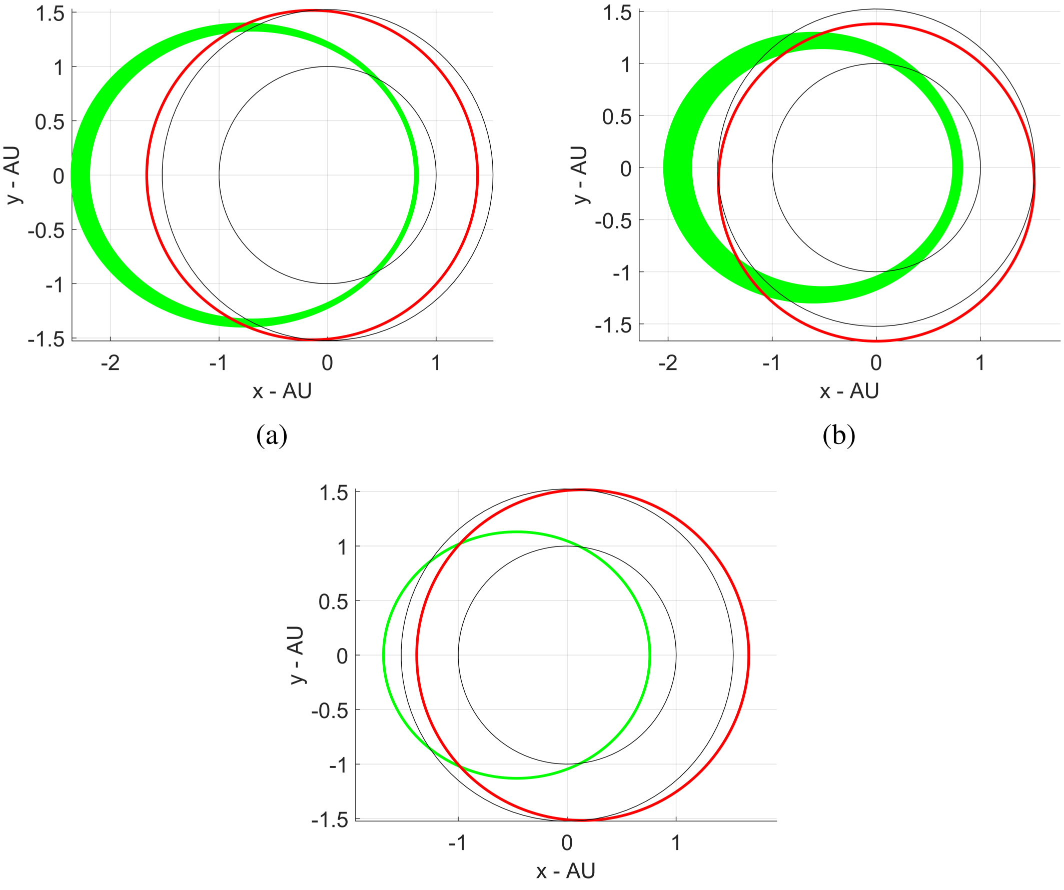
Similar effects arise from significant radial velocity differences in the encounter with Mars, having the highest orbital eccentricity among the planets considered. The sequence from Tisserand graph exploration (Algorithm 3) that has the closest infinity velocities to the solution obtained with nominal planetary ephemerides is shown in Table 4.9. A sensitivity analysis of the transfer orbits on the Mars-Earth leg is performed. The transfer starts at Mars with an infinity velocity of 10 km/s, and ends on Earth with an infinity velocity of 11 km/s. The closest time of flight on the Mars-Earth leg from a circular co-planar model is 508 days, that is far from the actual value of 649 from the nominal planetary ephemerides solution.

The orbital eccentricity of Mars is then restored (0.0934) and assumed to be coplanar to the one of the Earth. The information about this transfer orbit is known from Eq. 4.19, in terms of (α, v_∞). Small variations are applied to (α, v_∞) that range between -0.5 km/s and +0.5 km/s for the infinity velocity and -0.1 rad and +0.1 rad for α. Different transfer orbits are thus computed using Eq. 4.19. The time of flight between Mars and Earth is then computed. This process is repeated for different values of the argument of periapsis at Mars. Figure 4.19 shows Earth-Mars transfers for different argument of periapsis at Mars.

Table 4.9: Closest EVEMEJ sequence on Tisserand graph compared to the nominal planetary model solution.

Planet:	Earth	Venus	Earth	Mars	Earth	Jupiter
v_∞:	3 km/s	5 km/s	9 km/s	10 km/s	11 km/s	6 km/s

Applying such small variations on the information of the Tisserand graph corrects the values obtained from a circular and coplanar model by finding different orbits that can have the same departure condition on Mars. Modifying the argument of periapsis of



(a)

(b)

(c)

Figure 4.19: Representation of Mars-Earth transfers for different argument of periapsis, namely 0 rad (a), $\pi/2$ rad (b) and π rad (c). [181]

Mars' orbit highlights the impact of its eccentricity. As a result, a list of different and possible transfer times is obtained. The range of transfer time is between 419 days and 662 days, for the values mentioned above.

It should be noted that, while the variations of the infinity velocity and turn angle generate different values of time of flight, this is mostly the change in the argument of periapsis that broadens the range of possible time of flight. Figure 4.20 shows the evolution of the range of time of flight (due to the variations mentioned) with the argument of periapsis. The values obtained for the time of flight are within the interval. This proves that the difference in the transfer time observed is due to the differences between the model used for the Tisserand graphs and the reality.

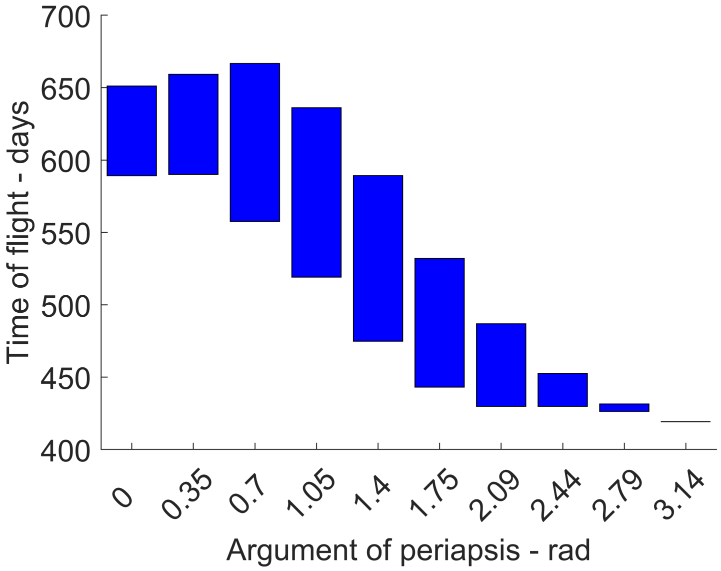


Figure 4.20: Evolution of the ranges of transfer time with the argument of periapsis of Mars. [181]

4.5 Conclusions

This Chapter presented the derivation of equations that fully describe the Tisserand graphs, assuming circular co-planar motion of the planets around the Sun. It has been shown how to exploit such graphs to construct MGA sequences that are useful for successive refinement in higher-fidelity models, also accounting for DSMs that allow infinity velocity leveraging at given bodies. The procedure allows to identify globally optimal paths for primary missions of interest.

Moreover, compared to existing literature on the subject, the exploitation of resonant loci has been applied to the preliminary assessment of high-inclination MGA tours, that are ideal for specific mission options such as the Jovian moon tour of JUICE, Solar Orbiter and Dolphin mission.

The benefits and limitations of Tisserand graphs have also been discussed. One can summarize the following points:

- Tisserand graphs provide very important information about the planetary sequences and infinity velocities required to reach specific orbital regions.

- Such information can be used in successive stages of design to substantially reduce the effort in the exploration of the search space, without losing information about global optimality and Pareto front structure, as proven in section 4.4.1.
- The information on the planetary phasing should be taken with care, as some solutions might be discarded in one model (e.g., circular coplanar) because large manoeuvres might be needed to achieve the correct phasing, as discussed in section 4.4.2.

Chapter 5

Multi-Objective MGA Design via Dynamic Programming

The second step of the overall framework consists in exploring the search space in terms of planets phasing and DSMs for sequences of planets that come from the Tisserand map analysis. The efficiency of this step lies in finding globally optimal Pareto fronts for any sequence identified. This is achieved by means of a multi-objective dynamic programming (MODP) algorithm that solves this task. However, to apply dynamic programming principles, the MGA problem needs to be transcribed into a multi-stage decision process, on which the optimization of a given sequence is performed sequentially, i.e., one planet-to-planet leg at a time.

This Chapter, after recalling the mathematical models employed (section 5.1), deals with this transcription process (section 5.2) and the way to efficiently explore the transcribed space (section 5.2.1). It also provides mathematical proof (section 5.4) on the connection to the full MGA-DSM model as from Chapter 3. Section 5.5 then provides relevant test cases and numerical examples to demonstrate the effectiveness of the proposed solution method. Section 5.6 then expands on the theory about resonant transfers from Chapter 4, by computing high-inclination MGA tours exploiting resonances at planetary encounters.

Finally, section 5.7 discusses the main achievements and conclusions.

5.1 Problem Definition

The MGA trajectory design is a global optimization problem in its nature. For a given planetary sequence, there exist several locally optimal trajectories in terms of planets phasing, thrusting arcs and fly-by parameters constituting a search space of complex configuration. Solving the MGA problem automatically, i.e., finding the planetary sequence and a trajectory that are optimal with respect to some mission-related criteria, corresponds to solve a MINLP/HOCP problem. In this case, planets are considered as 'targets' in the sense that visiting one planet or another enables to reach specific orbits or Solar System objects, that is one needs to target specific planets to make that orbit/object reachable. In such problems, there is a function $F(X,y)$ to be minimized, depending on both integer (X) and continuous-varying variables (y) (see also Chapter 2). The function $F(X,y)$ can either encode one single objective or a number n_{obj} of competing objectives to be optimized simultaneously. Following the definition of multi-objective optimization problem from Chapter 2, a general multi-objective MINLP presents the following structure:

$$\begin{aligned} \text{Minimize: } & F(X,y) = f_1(X,y),...,f_{n_{obj}}(X,y) \\ \text{Subject to: } & g_i(X,y) = 0, \forall i = 1,...,m_{eq} \\ & g_i(X,y) \geq 0, \forall i = m_{eq}+1,...,m_{in} \\ & X_{lb} \leq X \leq X_{ub} \\ & y_{lb} \leq y \leq y_{ub} \end{aligned} \tag{5.1}$$

where $g_i(X,y)$ represents the constraints of the problem at hand (e.g., overall mission duration or Δv), where m_{eq} and m_{in} are the cardinalities for equality and inequality constraints, respectively; (X_{lb},y_{lb}) and (X_{ub},y_{ub}) represent box constraints, i.e., lower and upper bounds for (X,y), respectively.

In the case of MGA trajectory optimization, the design variables include the sequence of planets to be visited, which is encoded in the integer vector set X, as well as the visiting epochs and other continuous variables which describe spacecraft manoeuvres, such as the fly-bys or DSMs, all of which will be encoded in the continuous-varying vector y. The functions $F(X,y)$ and $g_i(X,y)$ would then represent mission critical parameters such as the common Δv cost of the entire transfer and mission duration, but also other mission-specific objectives.

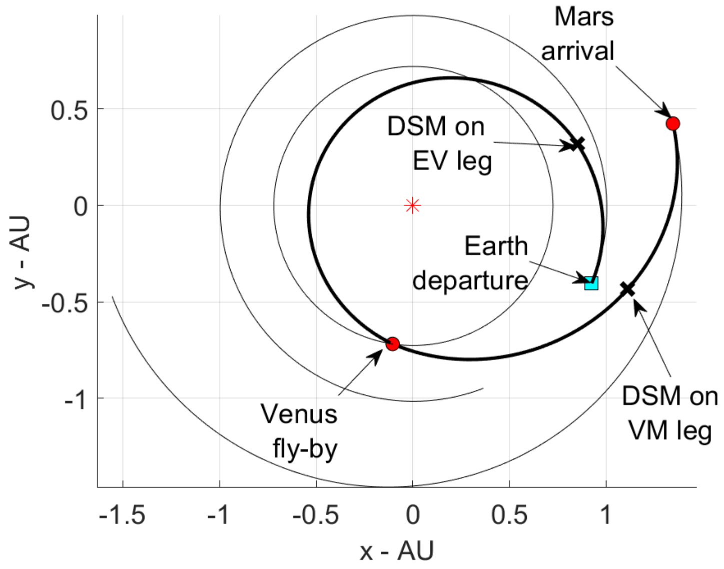


Figure 5.1: Example of an EVM trajectory with DSMs on both EV and VM legs.

Figure 5.1 shows an example trajectory which follows an Earth (E) – Venus (V) – Mars (M) sequence (EVM) with DSMs on both EV and EM legs. Table 5.1 provides a description of the integer and continuous variables involved in the problem at hand. In such example, vector X encodes a total of 3 objects (i.e., the planets), and vector y encodes 10 variables, defining all the events necessary to characterize the trajectories followed by the spacecraft between each planet. The model used here is the so-called MGA-DSM [128–131] that is also presented in Chapter 3, on which a DSM is assumed between two consecutive planetary encounters. In this model, a propagated arc in restricted two-body dynamics is assumed after each object encounter (either departing or swing-by planet) until the DSM position, followed by a Lambert arc between the DSM location and the next planetary encounter.

Table 5.1: Integer and continuous variables for the MINLP instance of the MGA problem with DSMs.

Integer variables (X)	**Description**
$X_i, \forall i = 1, \ldots, n_{int}$	Objects in the sequence (n_{int} is the number of integer variables)
Continuous variables (y)	**Description**
$[t_0, v_{\infty,dep}, u, v, T, \eta]_1$	For the first planet-to-planet leg: t_0 is the launch date; $v_{\infty,dep}$ is the spacecraft velocity relative to the departing body; (u,v) define the heliocentric direction of the spacecraft launch as per Chapter 3; T is the transfer time between two bodies; η is the time fraction at which a DSM is performed.
$[r_p, \zeta, T, \eta]_i, \forall i = 2, ..., n_{int} - 1$	For all the successive planet-to-planet legs: r_p and ζ are periapsis and inclination of the fly-by hyperbola, respectively.

One should notice that if either more complex transfer options or more complex dynamical frameworks are implemented, such as low thrust and ephemeris model, the number of optimizable parameters would rise sharply. Other strategies are also found in literature to model the transfers between two planetary encounters, such as multiple-shooting algorithm [25], primer vector theory [242], or the newly derived MGA-nDSM model (Chapter 3), used to mitigate some of the difficulties associated to the MGA-DSM model, mainly on the number of manoeuvres between two planetary encounters. However, for the purposes of the present Chapter, the MGA-DSM model is deemed sufficient to provide good representative trajectories of the MGA problem.

In this Chapter, we consider $F(X,y)$ being a function of two objectives to be optimized, i.e., $F(X,y) = (f_1(X,y), f_2(X,y))$, on which (similarly to Eq. 4.29):

$$\begin{cases} f_1 & = v_{\infty,dep} + \sum_{i=1}^{n_{int}-1} \Delta v_i + v_{\infty,arr} \\ f_2 & = \sum_{i=1}^{n_{int}-1} T_i \end{cases} \tag{5.2}$$

where $v_{\infty,dep}$ and $v_{\infty,arr}$ are the spacecraft velocities relative to the departing and arrival body, respectively, representing the manoeuvre to escape from the gravity of the first planet and the one to be captured within the last planet SOI [16], and Δv_i are the DSMs magnitude on each planet-to-planet leg of the transfer.

Solving the multi-objective optimization of MGA sequences as formulated in Eq. 5.1 and 5.2 with no a priori knowledge of the problem would only be feasible for formulations with very small search domains for both integer variable X and continuous-varying vector y, namely, either small fly-by sequences and/or launch windows, transfer times, etc. Hence, it is evident that solving the mixed-integer formulation of the MGA problem requires a process of refinement to manage this complexity efficiently.

To do so, the following pipeline is used, which is briefly assessed here:

- A criterion to select successive planetary encounters is conveniently used (this has been tackled in Chapter 4), mainly to assess the feasibility of different sequences in designing the mission at hand.
- A transcription of the problem from a mixed-integer formulation to a discrete optimization is useful (section 5.2) to explore the search space in an efficient manner. In this way, one exploits search space properties of the problem at hand and applies suitable graph-traversing techniques (sections 5.2.1 and 5.3).
- A refinement step is finally implemented (section 5.4) which takes each worthy solution identified with previous steps and finds all the relevant parameters of the original problem.

5.2 Transcription Using Defects

The Δv-defects model as described in Chapter 3 is here employed. This can be seen as an MGA-DSM model with $\eta_i = 0 \; \forall \; i = 1,...,n_{int}$ as from Table 5.1, i.e., on which the DSM occurs at each planetary encounter, i.e., at the fly-by epoch. Such model considers only a Lambert arc transfer for each object-to-object transfer in the sequence, thus only the times at each object are necessary. Velocity discontinuities between arriving and departing arcs, at a given object, are assumed equivalent to a DSM occurring at the encounter, see schematic in Figure 5.2. In other words, the magnitude of $\vec{v}_\infty^-$ is different from $\vec{v}_\infty^+$, and this difference is compensated by a Δv defect.

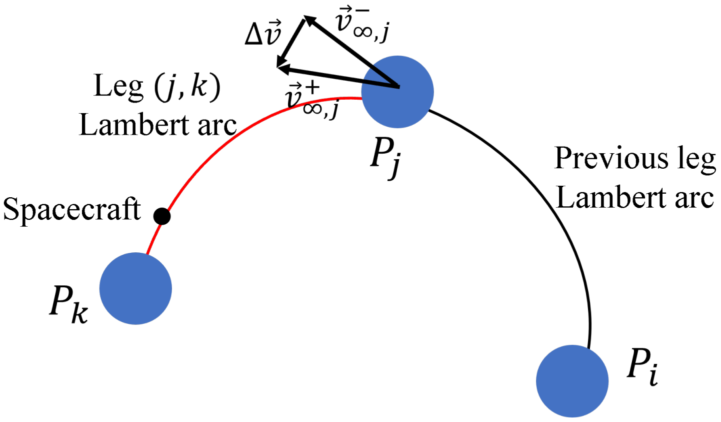


Figure 5.2: Sketch of spacecraft trajectory and Δv between point P_j and P_k.The spacecraft is moving from point P_j and P_k on the red track, having visited previously object P_i.

The point to highlight here is that the Δv-defects model allows to transcribe efficiently the complex search space of the MGA-DSM model as: (1) each planet-to-planet leg depends only upon the previously visited object through the vector $\vec{v}_\infty^-$; (2) the decision variables (i.e., the times at which each encounter occurs) are discretized and made varying over grids.

The problem is thus transcribed into a discrete optimization problem of finding the assignment of planetary visiting epochs that minimizes some user-defined functions (as the ones in Eq. 5.2). For example, considering a transfer from object a to d, with fly-bys at objects b and c, the optimization of the overall $[a,b,c,d]$ sequence is performed in successive stages:

1. Lambert problems are solved over a grid of departure dates and transfer durations for the first leg, i.e., $[a,b]$ [1].

[1] Izzo's algorithm is implemented for solving Lambert's problem. The interested reader is referred to

2. At the start of next leg, namely $[b,c]$, departure dates are updated using the arrival epochs at b from previous leg, plus range of duration for the current leg towards c. This step is repeated for all consecutive legs.

3. For each fly-by in the sequence, for all the incoming routes to a planet, that correspond to the different start and transfer times from the previous leg, the incoming relative velocity $\vec{v}_\infty^-$ is compared with the outgoing relative velocity $\vec{v}_\infty^+$ or routes to the next flyby (for matching arrival and departure times) and the defect is computed as from Eq. 3.10.

Tisserand-based information as from Chapter 4 also allows to identify the need for resonant orbits on legs that visit the same planet consecutively. These can occur when the time of flight is such that the spacecraft encounters the same point for two encounters. Such transfers are thus characterized by a ratio of integers between the planet and spacecraft orbit periods. The details of resonant orbits can be obtained analytically via derivation of post-fly-by relative velocity vector characteristics [153], thus solution of Lambert's problem is not required. For a given approach vector $\vec{v}_\infty^-$ then an infinite number of post fly-by resonant solutions exist, for the defined resonance, which differ in the inclination of the resonant orbit. This range of different inclination solutions can be retained for consideration at the next fly-by, or a baseline assumption of minimal inclination change can be assumed, leading to a simplification of the problem structure. More information on how to use resonant orbits to construct high-inclination tours can be found in section 5.6.

5.2.1 Graph Structure of the Search Space

This section describes the characteristics of the search space associated to the transcribed MGA problem described in previous section. Discrete problems are usually modeled with a search space made by grids of connected nodes. A common example is the Traveling Salesman Problem (TSP), where a salesman needs to visit a given number of cities, each representing a search node, which are connected by paths of a fixed length. As an optimization problem, the shortest path or tour around all cities is sought. Translating the TSP example to MGA missions, a spacecraft needs to fly-by several planets and each combination of planets will have an associated cost, usually Δv-driven. It should be noted that this analogy is only relevant for the transcribed MGA problem, which define visiting epochs that can vary discretely on grids.

As described in previous section and in Chapter 3, the cost of the path connecting two objects is associated with the Lambert arc connecting both objects at their respective encounter epochs. Figure 5.2 illustrates the spacecraft trajectory between two points P_j and P_k. Since P_j and P_k have each an associated encounter epochs $t^{enc}_{(.)}$, the time of flight between the two is uniquely defined (i.e., $T = t^{enc}_k - t^{enc}_j$) and, consequently, also the Lambert arc between these two points, for a given number of revolutions and energy solution. The spacecraft cost of connecting planet j and k at their respective visiting epochs is given by the impulsive manoeuvre Δv as from Eq. 3.10. Consequently, the cost of a given leg is not unique but depends upon the point prior to P_j, which will define the $\vec{v}^{-}_{\infty,j}$. Thus, to uniquely define the cost of a given leg between P_j and P_k, one needs to consider also the previously visited one, say P_i, so for the triplet (P_i, P_j, P_k), one has a unique cost. One should note that this optimal substructure property in the form of a triplet of individual nodes is common to all problems where fly-bys are to be considered (see also Chapter 6).

Because of this substructure of unique triplets, the search space can be modeled as a graph made by interconnected nodes. Each node encodes a couple of points with their encounter epochs. Following the example of previous section, on an overall $[a,b,c,d]$ sequence of objects, and their encounter epochs, say $[t_1,t_2,t_3,t_4]$, respectively, one has the following nodes $[A,B,C]$:

- $A = (at_1, bt_2)$
- $B = (bt_2, ct_3)$

Appendix A and [40]

- $C = (ct_3, dt_4)$

Each node thus encodes a trajectory linking two consecutive objects, being visited at the specified epochs. When connecting two consecutive nodes, the first object in the second node must be equal to the second object encoded in previous node. The cost of each connection between two nodes is then given by the Δv in Eq. 3.10. The first and last node have also a cost associated to $v_{\infty,dep}$ and $v_{\infty,arr}$, respectively. In this space, the cost between consecutive nodes is unique, which is the main advantage of modelling the search space in this way. Moreover, the problem is thus formulated in a way such that the solution can be seen as a combination of independent sub-problems, i.e., the transfers between the triplets of planets (P_i, P_j, P_k). Therefore, the sequence of objects has become a sequence of nodes, each node, i.e., planet-to-planet leg, being independent from the other ones. This allows dynamic programming techniques to be applicable to the problem at hand.

5.3 Graph Exploration via Dynamic Programming

Although the transcription process described in section 5.2 is similar among similar works [183–185, 188, 189], in the present paper the sub-structure of unique triplets as described in section 5.2.1 is usefully exploited to obtain globally optimal transfers on the transcribed search space via dynamic programming principles. The discrete problem of MGA missions is thus conveniently modeled as a tree-graph. Each node on the graph represents a transfer that can be incrementally constructed expanding one or more of its branches, i.e., adding a trajectory leg. In this way, the problem can be seen as a multi-stage decision process, on which the overall construction of MGA sequences is reached by making a series of lower-level choices, i.e., the selection of nodes between two different depth-level of the tree-graph. This also allows to better handle the constraints as from Eq. 5.1 (e.g., on maximum Δv defect, see also section 5.5), as branches that do not fit those boundaries are pruned out from the search space. Thus, two steps are necessary when expanding a tree-graph: (1) branching the nodes, (2) selecting the branched nodes to be kept for further expansion.

Among algorithms usually employed to scan tree-graphs, Depth First (DF) or Breadth First (BF) strategies are the most common [162]. These are known to be complete strategies, i.e., they allow to obtain the global optimum in discrete/combinatorial problems by keeping all the possible branched nodes while in the selection step. However, for practical space-related applications, this usually implies an infeasible number of trajectory branches to be evaluated and kept in memory. Beam Search (BS) algorithms might represent a very useful alternative [44], since the computational effort is bounded by heuristics that prevent the exploration of non-promising branches, thus only a limited number of nodes are kept at the selection step. For this reason, BS sacrifices the guarantee of global optimality in favor of computational efficiency.

5.3.1 Single- and Multi-Objective Dynamic Programming

Dynamic programming mitigates the computational burden associated to exhaustive DF/BF searches while guaranteeing global optimality for the problem at hand. To do so, dynamic programming exploits Bellman's principle of optimality to make optimal decision of nodes to be kept for further consideration at the selection step of the tree expansion. The Bellman's principle of optimality (in its single-objective formulation) states that regardless of the node at which the spacecraft currently is on the tree-graph, the optimal set containing this specific node would include the optimal subset of nodes before and after the visited one [116].

In other words, if the graph expansion happens to arrive at the same node at a specific tree-depth from different paths, then only the path with the minimum objective value is useful to be kept for further consideration. Figure 5.3 shows a representation of this principle. One notices that, at third depth-level, the tree expansion reaches node I from four different paths, namely: ACI, ADI, BEI, and BFI (recall that each node is made

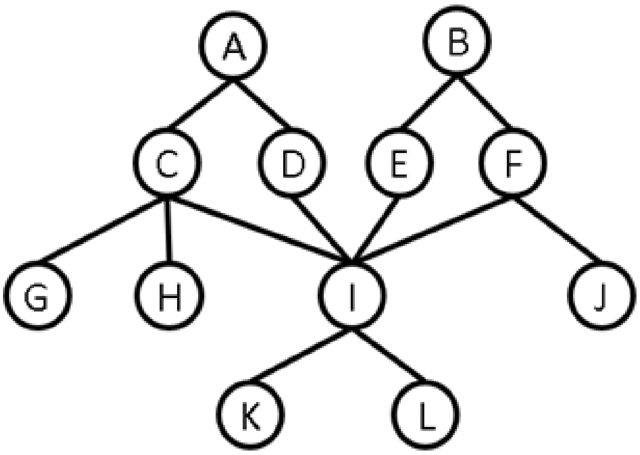
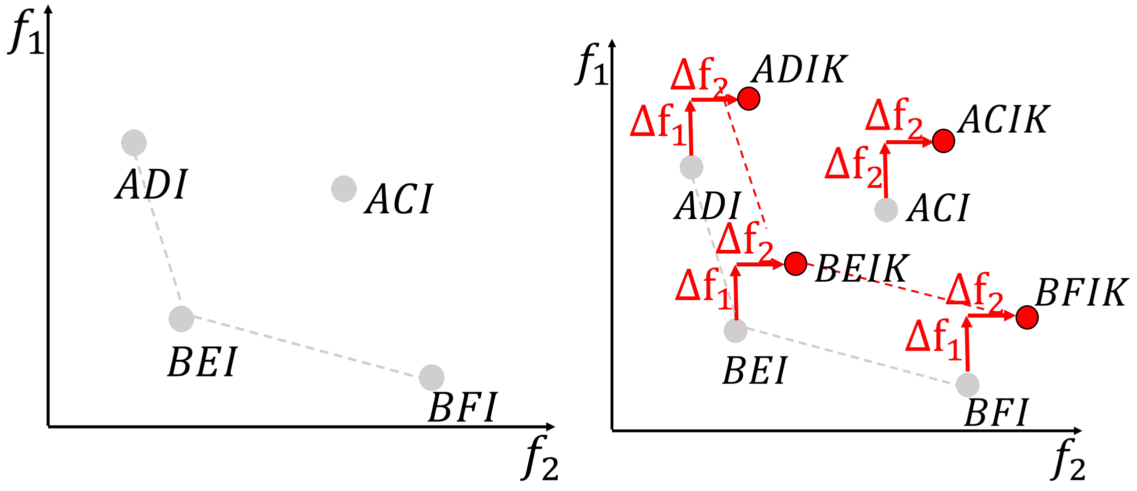


Figure 5.3: Tree-graph with common node at the third depth-level.

by a couple of objects and their visiting epochs as from section 2.a). Assume now that the cost of sequence BFI (e.g., f_1 or f_2 as from Eq. 5.2) is the lowest among all the other sequences that arrive in node I. Thus, any other node added after I (namely K and L) would imply an increase in cost that is the same for all the sequences, thus it is not needed to keep all ACI, ADI and BEI for further consideration, since they will always be worse than dominated by BFI in terms of cost function for any successive node added. On the other hand, all the sequences that do not have any node in common at the given tree-depth, namely ACG, ACH and BFJ, are kept for further expansion alongside BFI. This ultimately allows a sensible reduction of the number of paths that need to be kept in memory when exploring the tree. Therefore, in discrete optimization problems, single-objective dynamic programming (SODP) allows to identify in an automatic manner the sequence that minimizes a specific objective with the lowest number of paths to be stored in memory.

The Bellman's principle of optimality can be also extended to handle multi-objective optimization. The extended principle states that, regardless of the node at which the spacecraft currently is on the tree-graph, the Pareto-optimal set containing this specific node would include the Pareto-optimal subset of nodes before and after the visited one. Analogously to the SODP case, if multiple sequences arrive at the same node at a given



(a) (b)

Figure 5.4: Representation of different paths arriving to the same node at a specific tree-depth in the f_1, f_2 plane (a) and effect of adding a node to the same sequences (b). Dotted lines link nodes on the Pareto front.

tree-depth, then only the paths belonging to the Pareto front are useful and thus need to be kept for further consideration.

This can be seen intuitively following the same example of Figure 5.3. All the sequences that share the same node at a given tree-depth (i.e., the node I at the third level in this case), can be represented in a space that has a number of main axes as many as objective functions $F(X,y) = f_1(X,y),...,f_{n_{obj}}(X,y)$ in Eq. 5.1. For the sake of simplicity, let's consider just two objectives f_1 and f_2 (e.g., as from Eq. 5.2). The representation of the nodes in such space is given in Figure 5.4.a. As an example, one identifies a Pareto front with sequences ADI, BEI and BFI, while ACI is the dominated sequence. Adding any node to these sequences, e.g., node K, would imply a variation in all the objectives, namely Δf_1 and Δf_2 in Figure 5.4.b, which is the same for all the sequences. The Pareto front is thus preserved for the sequences ADIK, BEIK and BFIK, and any sequence dominated before adding node K is still dominated by the addition of this node, and thus is not needed for further expansion. In this way, analogously to the SODP case, in discrete optimization problems, multi-objective dynamic programming (MODP) allows to identify in an automatic manner the optimal Pareto front with the lowest number of paths to be stored in memory.

When expanding the tree-graph, one can either: (1) generate the list of feasible planetary sequence by means of Tisserand-based information as from Chapter 4 and then apply the tree expansion on each of them (explicit variant), or (2) exploiting the same Tisserand information directly at the tree-expansion step (implicit variant). One notices that the two options come with the same computational effort in terms of Lambert arcs solved and defects computed. In this work, the tree-graph expansion branches new trajectory legs only if the sequence is within a preloaded list of sequences (which is the result of the Tisserand exploration as from Chapter 4). One should notice that since the multi-objective optimization is performed on the transcribed space for each sequence in the list and such transcription is an approximation likely to have worse Δv of the successive refinement step as from section 5.4, it is important to keep all the different MGA sequences that arrive at a common node. Then, the leveraging is assessed in the refinement step for a correct trade off analysis, as different sequences can perform in different ways in the refinement step (see also section 5.4).

5.4 Defects Removal by DSM Correction Evaluation via State Transition Matrix

By applying the transcription process described in section 5.2, the problem is in practice decomposed into two consecutive sub-problems: firstly, the multi-objective discrete optimization, which aims at identifying promising MGA paths with respect to competing mission criteria, and secondly, a refinement step aiming at optimizing the continuous design variables given a fixed sequence. The key aspect of the proposed method is assessing the relationship between manoeuvres from the MGA-DSM model and the defects model. This section focuses on this relationship, and it shows the robustness of the proposed approach (i.e., transcription + SODP/MODP application) in representing mission scenarios that are easily convertible into higher-fidelity models (e.g., the MGA-DSM one).

Insertion of a DSM to remove an infinity velocity defect at the next fly-by is here used to establish the relationship between the manoeuvre and the corresponding correction of the defect. The incoming defect dependency on a preceding mid-course DSM is thus obtained. This dependency is referred to as leveraging ratio, i.e., the ratio between the defect and a precedent DSM. A crucial consideration is the maintenance of the subsequent rendezvous with the target planet for the fly-by, as well as the removal of the defect.

The DSM is assumed to take place at a time $t_0 = \eta T$ on a planet-to-planet leg, while the successive planetary encounter occurs at t_f. The DSM is derived in a reference frame which has $\hat{v}_v$, $\hat{v}_p$ and $\hat{v}_n$ as unit vectors, which are components along the velocity vector, in-orbit plane perpendicular to the velocity vector and out-of-plane normal, respectively. The DSM has an impact on the position vector achieved at the epoch of the fly-by. Therefore, a constraint vector $\vec{C}$ should be considered that maintains the relative position error with respect to the swing-by planet and the infinity velocity defect at zero. The constraints are thus $\vec{C} = [\vec{r}_{rel}, \Delta v]^T$, where $\vec{r}_{rel} = \vec{r} - \vec{r}_{pl}$ is the difference between the spacecraft and planet position vectors at t_f (i.e., $\vec{r}$ and $\vec{r}_{pl}$, respectively), and Δv is the infinity velocity defect magnitude at t_f. The control is $\vec{U} = [\Delta\vec{v}(t_0), t_f]^T$, on which $\Delta v(t_0) = [DSM_v, DSM_p, DSM_n]^T$ is the manoeuvre vector, written in the reference frame identified by $\hat{v}_v$, $\hat{v}_p$ and $\hat{v}_n$ as defined above.

The required change in the constraint vector is $\begin{bmatrix} 0 & 0 & 0 & -\Delta v \end{bmatrix}^T$. The increment in the control $\Delta\vec{U}$ is found approximately from a single Newton-Raphson like iteration for $\vec{C} = 0$ by:

$$\frac{\partial \vec{C}}{\partial \vec{U}} \Delta\vec{U} = \begin{bmatrix} 0 & 0 & 0 & -\Delta v \end{bmatrix}^T \tag{5.3}$$

From which the control increment $\Delta\vec{U}$ is found via matrix inversion. From Eq. 5.3, the vector $[0,0,0,-\Delta v_\infty]^T$ corresponds to the required change in the constraint vector $\vec{C}$, and the matrix $\frac{\partial \vec{C}}{\partial \vec{U}}$ is defined as follows:

$$\frac{\partial \vec{C}}{\partial \vec{U}} = \begin{bmatrix} \frac{\partial \vec{r}_{rel}(t_f)}{\partial \Delta\vec{v}(t_0)} & \frac{\partial \vec{r}_{rel}(t_f)}{\partial t_f} \\ \frac{\partial |\vec{v}_{rel}|(t_f)}{\partial \Delta\vec{v}(t_0)} & 0 \end{bmatrix} \tag{5.4}$$

where $\vec{v}_{rel} = \vec{v} - \vec{v}_{pl}$ is the spacecraft velocity vector relative to the fly-by planet computed at time t_f ($\vec{v}$ and $\vec{v}_{pl}$ are he spacecraft and planet velocities at t_f, respectively).

By computing the State Transition Matrix (STM) Φ between t_0 and t_f one has:

$$\Phi = \begin{bmatrix} \frac{\partial \vec{r}(t_f)}{\partial \vec{r}(t_0)} & \frac{\partial \vec{r}(t_f)}{\partial \vec{v}(t_0)} \\ \frac{\partial \vec{v}(t_f)}{\partial \vec{r}(t_0)} & \frac{\partial \vec{v}(t_f)}{\partial \vec{v}(t_0)} \end{bmatrix} \tag{5.5}$$

Using $\frac{\partial \vec{r}(t_f)}{\partial \vec{v}(t_0)} = \frac{\partial \vec{r}_{rel}(t_f)}{\partial \Delta\vec{v}(t_0)}$, then:

$$\frac{\partial \vec{r}_{rel}(t_f)}{\partial \Delta\vec{v}(t_0)} = \begin{bmatrix} \frac{\partial \vec{r}(t_f)}{\partial \vec{v}(t_0)}\hat{v}_v & \frac{\partial \vec{r}(t_f)}{\partial \vec{v}(t_0)}\hat{v}_p & \frac{\partial \vec{r}(t_f)}{\partial \vec{v}(t_0)}\hat{v}_n \end{bmatrix} \tag{5.6}$$

Moreover, one has:

$$\frac{\partial \vec{r}_{rel}(t_f)}{\partial t_f} = \vec{v}(t_f) - \vec{v}_{pl}(t_f) \tag{5.7}$$

where $\vec{v}(t_f)$ and $\vec{v}_{pl}(t_f)$ are again the spacecraft and planet velocities computed at t_f, respectively.

Then, to evaluate the following:

$$\frac{\partial |\vec{v}_{rel}|(t_f)}{\partial \Delta\vec{v}(t_0)} = \begin{bmatrix} \frac{\partial |\vec{v}_{rel}|(t_f)}{\partial DSM_v} & \frac{\partial |\vec{v}_{rel}|(t_f)}{\partial DSM_p} & \frac{\partial |\vec{v}_{rel}|(t_f)}{\partial DSM_n} \end{bmatrix} \tag{5.8}$$

One uses:

$$\frac{\partial |\vec{v}_{rel}|(t_f)}{\partial DSM_i} = \frac{\partial (\vec{v}(t_f) - \vec{v}_{pl}(t_f))}{\partial DSM_i} \cdot \hat{v}_{rel} = \frac{\partial \vec{v}(t_f)}{\partial DSM_i} \cdot \hat{v}_{rel} \tag{5.9}$$

on which $\frac{\partial \vec{v}(t_f)}{\partial DSM_i} = \frac{\partial \vec{v}(t_f)}{\partial \vec{v}(t_0)} \cdot \hat{v}_i$, for each $i = v, p, n$.

Therefore, it is possible to compute $\frac{\partial \vec{C}}{\partial \vec{U}}$ as defined in Eq. 5.4 using Eq. 5.6, 5.7 ans 5.8, and thus $\Delta\vec{U}$ from Eq. 5.3. This is again evaluated via STM along the nominal trajectory for different values of t_0. The leveraging ratio $\Delta v_\infty / |\Delta\vec{v}|$ can also be derived as a function of t_0. Its maximum value, corresponding to the minimum $|\Delta\vec{v}|$ over the trajectory, can then be obtained and used to inform successive refinement stages on the position of the midcourse DSMs. The terms $\frac{\partial \vec{r}(t_f)}{\partial \vec{v}(t_0)}$ and $\frac{\partial \vec{v}(t_f)}{\partial \vec{v}(t_0)}$ used in Eq. 5.6 and 5.9 are obtained from the standard STM as in Eq. 5.5 for the trajectory between t_0 and t_f.

Hence, the refinement process takes all or a given subset of solutions from the grid optimization and reconstructs the fly-by parameters and midcourse manoeuvres as in the MGA-DSM model with the guess on: (1) departing dates and transfer times provided by the grid optimization and (2) optimal location of the manoeuvres with the help of the analytical procedure on the described with Eq. 5.3 to 5.9. Thus, the Δv defects can be replaced with DSMs occurring after a fraction of the transfer time between two consecutive swing-bys, according to the MGA-DSM model. It is important to note that the defects solutions are not approximations of the complete problem, but they solve the same fitness function $f_i(x,y)$ as for the refinement process. The visiting epochs and planetary fly-by parameters, variables encoded in vector y, identified in the defect model, only need to be refined in case a lower Δv solution exists in the same neighborhood. This is done via the above analysis, that checks if a variation in the visiting epochs exists such that a lower Δv solution exists if midcourse DSM is included to remove the defects. The refinement process in the MGA-DSM model is carried by re-optimizing the solutions using a PSO. Departing epoch in y vector is allowed to vary on a ± 30 days-range with respect to the corresponding grid optimization values, while visiting epochs on a ± 15 % range of transfer time of the leg.

For this process to be efficient, defects solutions must be close enough to the real minimum solution for the refinement step. This is shown in Figure 5.5, representing on the

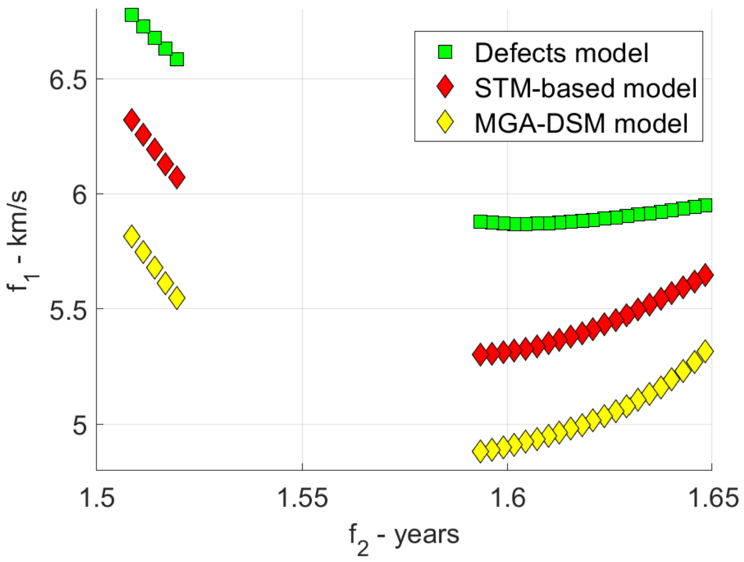


Figure 5.5: f_1 and f_2 values for different EVV trajectories arriving at the node (V,t_1,V,t_2) computed with models of different fidelity.

(f_1, f_2) plane different trajectories on an overall EVV mission that arrive at the same node. Recall that a node in the MGA graph transcription is made by a couple of objects (VV in this case) and their encounter epochs ($t_1 = -589$ MJD2000 and $t_2 = -183$ MJD200, in this example). One appreciates that all the solutions incoming to a given node belong to a single-funnel structure [248]. Therefore, the SODP/MODP selection process as from section 5.2 is robust as it allows to capture all the different funnels of the MGA problem not losing information in the transcription process, and such funnels are then efficiently refined within higher-fidelity models since no multi-funnels are found for a single node under evaluation.

Figure 5.5 also provides an illustration of the potential value obtained from using STM-based predictions to the MGA-DSM model. In particular, STM-based solutions are accurate in the sense that they provide a better estimate of the solutions in higher-fidelity models (average offset between STM-based model and defects model is about 400 m/s) as well as the representation of the overall funnel. This suggests that including the procedure as in Eq. 5.3 to 5.9 while in the branching step as from section 5.2 could be beneficial for including DSMs that lie closer to the fully optimized solutions, at the price of relatively small increase in computational cost due to STM evaluation at each node. One should also notice that defects solutions and STM-based solutions converge to the same MGA-DSM solutions, so including STM-based approximation is a compromise between computational effort and solution quality and depends upon the mission application.

5.5 Numerical Results and Discussion

Applications of the pipeline presented in Chapter 4 and sections 5.2, 5.3 and 5.4 are here discussed. These follow a logic of increased complexity and are briefly introduced here:

1. A transfer towards Saturn is optimized with respect to the two objectives f_1 and f_2 as in Eq. 5.2, assuming the sequence from Earth to Saturn to be known, i.e., EVVEJS. This is like the Cassini and so-called Cassini-2 design problem proposed by ESA [2]. However, compared to most of the literature on the same problem, here the multi-objective optimization is tackled. One tries to explore the launch window to find suitable swing-by dates for the proposed sequence, as well as DSMs. This is done to assess the ability of the proposed pipeline in identifying Pareto-optimal paths (section 5.5.1) and to test to test the efficiency of using the transcription process alongside SODP/MODP approaches when exploring the transcribed search space (section 5.5.2). An analysis on the relationship between thrust model types (i.e., defects and DSMs) is included to prove the robustness of the methodology in representing primary missions of interest (section 5.5.3).

2. One now assumes that the sequence reaching Saturn is not known but needs to be selected as part of the multi-objective optimization process (section 5.5.4). It will be shown that it is possible to identify sequences that are competitive with respect to the well-known EVVEJS and that contribute to the Pareto front. To do so, a much wider exploration with respect to current literature is performed, both in terms of

[2] `https://www.esa.int/gsp/ACT/projects/gtop/` last accessed March 2022

transfer times between planets and number of revolutions between two consecutive encounters.

3. Novel transfer options are explored in the context of a sample return mission towards comet 67P/Churyumov-Gerasimenko, same target as Rosetta mission [249] (section 5.5.5). The increased complexity of this mission scenario lies on: the extended launch window (10 years are considered in the 2030-2040 time-frame) and the high number of planetary encounters (sequence with up to 10 objects are found).

The following discussion highlights the advantages of the approach presented in this Chapter, which are:

1. Fly-by sequences, departing dates, transfer times, number of revolutions around the Sun and manoeuvres size and location do not need to be known a priori.
2. Search spaces are substantially larger than those presented in similar problems in current literature.
3. Optimal trajectories with respect to competing mission objectives in an overall multi-objective optimization can be obtained in an efficient and robust manner, also showing novel transfers with respect to literature.

5.5.1 Multi-Objective Optimization of EVVEJS Sequence

Here, the multi-objective optimization of sequence EVVEJS is assessed within the 1997 launch window. This is the same sequence employed by Cassini mission [171], which was launched in October 1997 and arrived at Saturn in 2004. Cassini required at least one large DSM between two Venus fly-bys to reach its destination. This problem is only barely assessed in literature as usually only the single-objective optimization is considered. Some multi-objective problems have been formulated [201–203], but typically on smaller search spaces than the one considered here.

Table 5.2: Optimization scenario for EVVEJS sequence for a launch in 1997.

Design variables	Values and bounds
Sequence	Known: EVVEJS
Departure velocity magnitude	$v_{\infty,dep} \in [3,5]$ km/s
Maximum defect at each fly-by	$\Delta v \in [0,2]$ km/s
Launch window	$t_0 \in [-1095.5, -730.25]$ MJD2000
Number of revolutions about the Sun	$N_{rev} \leq 1$
Transfer times between planets	$T_1 \in [30,400]$ days $T_2 \in [100,470]$ days $T_3 \in [30,400]$ days $T_4 \in [400,2000]$ days $T_5 \in [1000,6000]$ days

The optimization is performed on the transcribed space as from section 5.2. In this case, the Δv computed as in Eq. 3.10 may lead to larger values. This is because they are manoeuvres applied immediately after departing from a fly-by, thus not representing DSMs in a real mission design context. In this case, a simple post-processing step as described in 5.4 is needed and again proves the robustness of the method in converging to benchmark solutions.

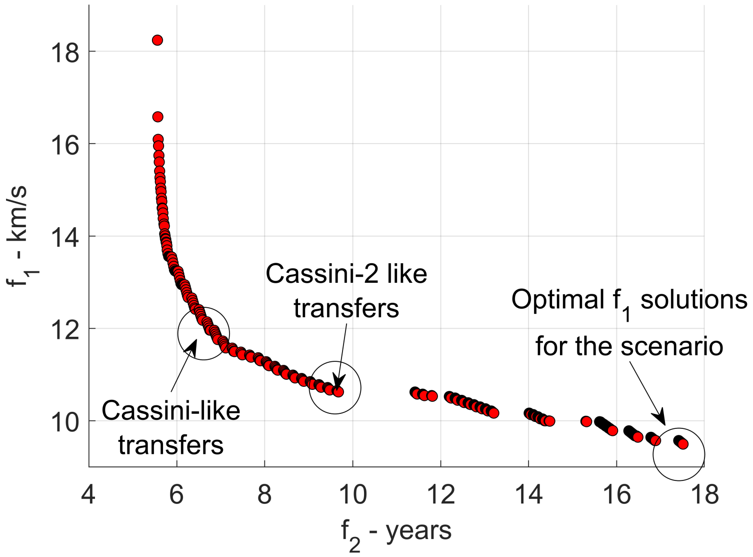


Figure 5.6: Pareto front of f_1 and f_2 objectives as from MODP optimization for EVVEJS. Primary missions of interest for the scenario are highlighted.

The problem to be solved consists in finding the optimal assignment of departing date and visiting epochs for all the planets in the sequence minimizing the objectives f_1 and f_2 as in Eq. 5.2. The optimization scenario is described in Table 5.2 for a launch window in 1997. One should notice that the transfer time bounds are chosen from available literature for bench-marking [3] [131]. For the grid optimization, step size in start date and durations, st_s and st_d, respectively, are chosen to be $st_s = st_d = 3$ days for EVVE sub-sequence evaluation and $st_d = 6$ days for EJS sub-sequence. Larger step sizes are admissible for transfer to outermost planets since these are less sensitive to grid sizes, due to their increased orbital periods [189].

Table 5.3: Results for Cassini-like mission compared to solutions from defect model and refinement. When a manoeuvre is not present between two planetary encounters, a '$--$' is included.

Event	**Cassini [171]**	**Defects solution**	**Refined solution**
Earth departure	Oct. 6, 1997	Nov. 23, 1997	Oct. 20, 1997
$v_{\infty,dep}$	4.25 km/s	4.01 km/s	4.00 km/s
Δv_1	$--$	$--$	$--$
Venus fly-by	Apr. 21, 1998	May 22, 1998	Apr. 29, 1998
Δv_2	0.466 km/s	1.97 km/s	0.431 km/s
Venus fly-by	Jun. 20, 1999	Jul. 01, 1999	Jun. 26, 1999
Δv_3	$--$	0.604 km/s	$--$
Earth fly-by	Aug. 19, 1999	Aug. 20, 1999	Aug. 18, 1999
Δv_4	$--$	0.214 km/s	$--$
Jupiter fly-by	Dec. 30, 2000	Jan. 01, 2001	Jan. 11, 2001
Δv_5	$--$	$--$	$--$
Saturn arrival	Jul. 01, 2004	Jun. 06, 2004	Sep. 19, 2004
$v_{\infty,arr}$	5.59 km/s	5.47 km/s	5.17 km/s
f_1	10.3 km/s	12.3 km/s	9.60 km/s
f_2	6.73 years	6.61 years	6.91 years

The effect of st_s and st_d alongside the maximum defect admissible at each fly-by is crucial when assessing the efficiency of the proposed approach and it is always a compromise between solutions quality and computational effort (see later section 5.5.2). This setup allows to obtain all the missions of interest for the given transfer scenario on a wide Pareto front, ranging from less than 6 years to almost 18 years of transfer time, as shown in Figure 5.6. Specifically, the Pareto front is comprehensive in that it correctly identifies the primary mission of interest, which are: (1) the actual Cassini mission, i.e., a fast transfer to Saturn of about 7 years; (2) the Cassini-2 problem solutions with relaxed time constraints, i.e., solutions with about 9.9 years, corresponding to the best-known time constrained solution; (3) optimal solutions for the given transfer scenario with transfer

[3] `https://www.esa.int/gsp/ACT/projects/gtop/`, last accessed March 2022

duration of about 17.3 years.

Table 5.4: Results for Cassini-2 like mission compared to solutions from defect model and refinement.

Event	**Cassini-2** [a]	**Defects solution**	**Refined solution**
Earth departure	Nov. 13, 1997	Nov. 23, 1997	Nov. 11, 1997
$v_{\infty,dep}$	3.26 km/s	3.88 km/s	3.30 km/s
Δv_1	0.480 km/s	−−	0.462 km/s
Venus fly-by	Apr. 29, 1998	May 20, 1998	Apr. 30, 1998
Δv_2	0.398 km/s	1.83 km/s	0.398 km/s
Venus fly-by	Jun. 27, 1999	Jul. 02, 1999	Jun. 28, 1999
Δv_3	−−	0.682 km/s	−−
Earth fly-by	Aug. 20, 1999	Aug. 21, 1999	Aug. 20, 1999
Δv_4	−−	−−	−−
Jupiter fly-by	Mar. 31, 2001	Apr. 18, 2001	Apr. 01, 2001
Δv_5	−−	−−	−−
Saturn arrival	Apr. 09, 2007	Apr. 24, 2007	Apr. 04, 2007
$v_{\infty,arr}$	4.24 km/s	4.21 km/s	4.24 km/s
f_1	8.38 km/s	10.6 km/s	8.40 km/s
f_2	9.40 years	9.41 years	9.39 years

[a] See footnote 2

Details and trajectory representations can be found in Tables 5.3, 5.4 and 5.5, as well as in Figure 5.7, 5.8 and 5.9, for both Cassini, Cassini-2 and optimal solution sequences, respectively. The solutions identified correspond closely within few days to referenced solutions. One notices that to the best knowledge of the author no reference solutions exist for the optimal sequences in terms of Δv consumption (Table 5.5 and Figure 5.9), again proving the efficiency of the pipeline in comprehensively solving the multi-objective optimization.

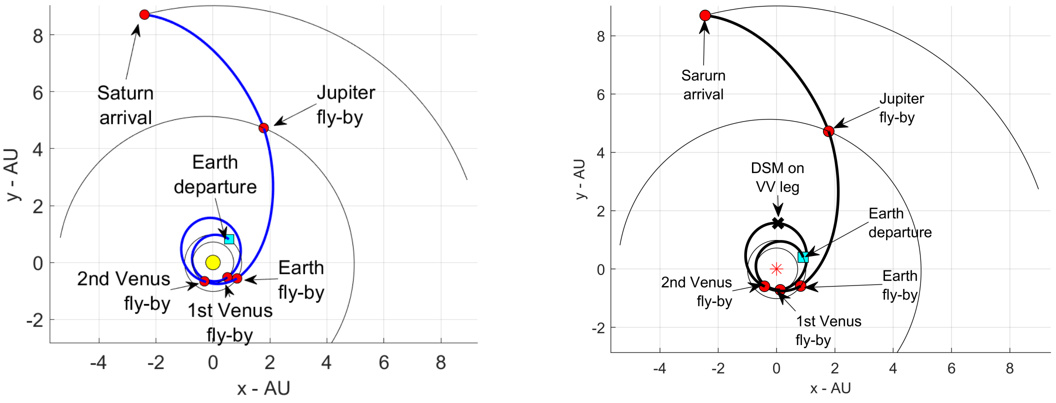
Table 5.5: Results for optimum solution in the given mission scenario compared to solutions from defect model and refinement.

Event	**Defects solution**	**Refined solution**
Earth departure	Nov. 13, 1997	Nov. 10, 1997
$v_{\infty,dep}$	3.63 km/s	3.59 km/s
Δv_1	−−	0.694 km/s
Venus fly-by	May 10, 1998	May 01, 1998
Δv_2	1.16 km/s	0.180 km/s
Venus fly-by	Jun. 02, 1999	Jun. 29, 1999
Δv_3	0.0933 km/s	−−
Earth fly-by	Aug. 27, 1999	Aug. 26, 1999
Δv_4	0.139 km/s	−−

Continued on next page

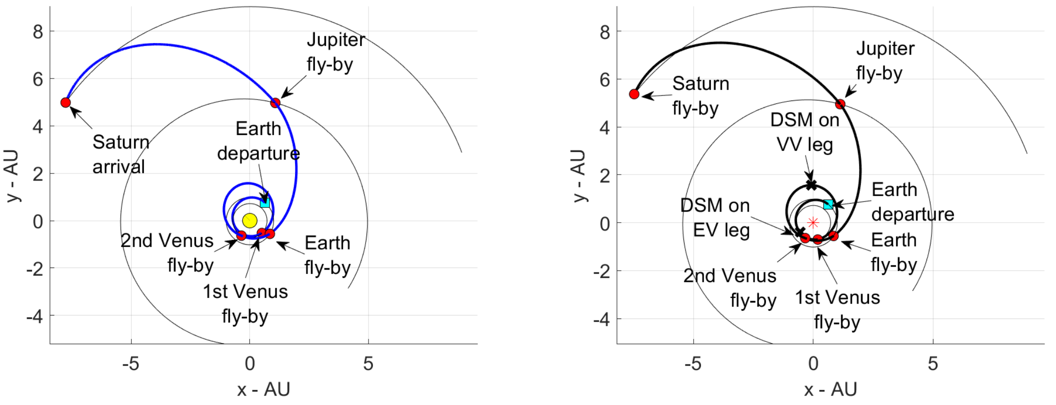
Table 5.5 – *Continued from previous page*

Jupiter fly-by	Jun. 18, 2002	Jun. 12, 2002
Δv_5	$--$	1.70 km/s
Saturn arrival	Mar. 02, 2015	Feb. 05, 2017
$v_{\infty,arr}$	4.27 km/s	2.08 km/s
f_1	9.29 km/s	7.55 km/s
f_2	9.30 years	19.2 years



(a) (b)

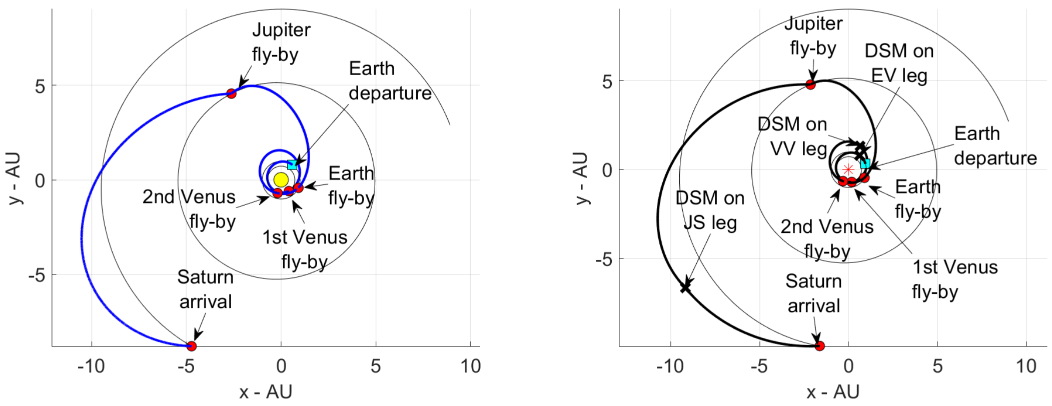
Figure 5.7: Cassini-like EVVEJS transfer with departure date in 1997 as resulting before (a) and after (b) the refinement process.



(a) (b)

Figure 5.8: Cassini-2 EVVEJS transfer with departure date in 1997 as resulting before (a) and after (b) the refinement process.

From Table 5.3, the nominal Cassini transfer lasts under 7 years from launch to rendezvous with Saturn. As a fast and time-constrained transfer, this mission does not represent the optimum in terms of f_1 objective. Faster transfers increase the infinity velocity at Saturn and lead to increased transfer velocity at the Earth, as well as larger DSMs.



(a) (b)

Figure 5.9: Optimal solution for EVVEJS transfer scenario with departure date in 1997 as resulting before (a) and after (b) the refinement process.

From Table 5.4, the best reported solution for the Saturn problem at this launch year (i.e, Cassini-2), as compared to the actual Cassini one, is characterized by lower infinity velocities at departure and arrival, thus better f_1 objective, but with much higher transfer time, i.e., higher f_2, mainly due to the last Jupiter-Saturn leg of the transfer. However, Pareto front in Figure 5.6 suggests that this solution is still constrained in terms of duration and significant improvements in terms of f_1 objective can be obtained. This is confirmed from Table 5.5, where the infinity velocity at Saturn is leveraged by a large DSM on the last leg, at the price of increased f_2 value (see also later section 5.5.3).

5.5.2 Assessment of Single-/Multi-Objective Dynamic Programming on MGA Trajectory Optimization

In multi-objective optimization of the transcribed MGA trajectory design as shown in 5.2, a full evaluation (FE) of all possible combinations of departing dates and transfer times in a sequence of legs linked by gravity assist usually makes the number of possible routes to rise exponentially with the number of legs. This issue is mitigated via SODP and MODP approaches as described in section 5.3. The key question is how feasible FE, SODP and MODP are when executed on a typical laptop. This is answered here by assessing the stages of evaluation and numbers of evaluation per stage for the EVVEJS example considered in Table 5.2, which are as follows:

1. The number of Lambert arcs evaluated (N_L)

2. The number of defects evaluated (N_d)

3. The number of routes for FE ($N_{r,FE}$), SODP ($N_{r,SODP}$) and MODP ($N_{r,MODP}$)

The rate of rise in N_L, N_d, $N_{r,FE}$, $N_{r,SODP}$ and $N_{r,MODP}$ depends critically on the discretization considered in evaluating each leg (i.e., step size in start st_s and duration st_d), and on the non-linear constraints applied, generally on maximum Δv defect at each encounter. A parametric study is thus performed to assess the feasibility of FE, SODP and MODP with respect to the step size for start date and durations, st_s and st_d, respectively, and the maximum defect Δv_{max}. Table 5.6 highlights the cases considered here. It should be noted again that larger step sizes are used for EJS legs as transfers to outermost planets are less sensitive to grid sizes due to the increased orbital periods [189].

Table 5.6: Values used in parametric study for multi-objective optimization of EVVEJS.

	st_s **[days]**	st_d **[days] (EVVE)**	st_d **[days] (EJS)**
Case 1	2	2	4
Case 2	3	3	6
Case 3	5	5	10

Regarding the maximum defect limit Δv_{max}, this is linked to limitations on spacecraft propulsion system, thus large values are not feasible for standard spacecraft designs. To achieve computational efficiency, Δv defects should be the minimum possible, but subject to the condition that locally optimal solutions are not lost, and the Pareto sets characteristics are retained. This implies an upper limit on the DSM in-between two consecutive swing-bys, which is related by the leveraging ration, i.e. the ratio between Δv defect and precedent DSM. As a general consideration, one should expect degradation of the Pareto front characteristics with larger step sizes and lower Δv_{max}. One should highlight that the Case 2 from Table 5.6 alongside $\Delta v_{max} = 2$ km/s employed in the numerical study in section 5.5.1 is able to capture wide Pareto front and primary missions of interest for the optimization scenario considered. Table 5.7 represents the computational effort in terms of N_L, N_d, $N_{r,FE}$, $N_{r,SODP}$ and $N_{r,MODP}$ with respect to the case considered. The number of Lambert problems solved N_L and the number of defects N_d decreases when coarser grids are considered. As it can be seen, although the number of defects and Lambert problems remain the same for both SODP, MODP and FE, the number of different MGA paths evaluated and stored in memory by each of the methods is substantially lower for both SODP and MODP when compared to FE. This can also be seen from Figure 5.10 explicitly showing an example for EVVEJS tree exploration using SODP. Two different sequences arrive at common node (V,t_7,E,t_8) at level 3, but only the bold one is kept for further expansion since the accumulated Δv is lower. The same happens at level 4, where the node (E,t_8,J,t_{10}) is reached by three routes, but only one is kept for further consideration.

Table 5.7: Computational effort for SODP, MODP and FE. No defects are computed on the first leg of the transfer, thus a '$--$' is included. The $\Delta v_{max} = 2$ km/s.

Case 1

	N_L	N_d	$N_{r,SODP}$	$N_{r,MODP}$	$N_{r,FE}$
EV	34038	$--$	2249	2249	2249
VV	14880	418314	1462	10924	26263
VE	11718	271932	4764	66249	1190503
EJ	64160	1910364	1772	28170	4864556
JS	670536	2216772	278910	6145707	2754878045
Totals	**795332**	**4817382**	**278910**	**6145707**	**2754878045**

Number of points in Pareto front **560**
Optimum for f_1 **9.447 km/s**

Case 2

	N_L	N_d	$N_{r,SODP}$	$N_{r,MODP}$	$N_{r,FE}$
EV	15128	$--$	995	995	995
VV	6572	123380	643	3403	7869
VE	5208	79732	2105	20302	240687
EJ	28035	562035	780	8479	654155
JS	292734	650520	121213	1754661	246966060
Totals	**347677**	**1415667**	**121213**	**1754661**	**246966060**

Number of points in Pareto front **333**
Optimum for f_1 **9.494 km/s km/s**

Case 3

	N_L	N_d	$N_{r,SODP}$	$N_{r,MODP}$	$N_{r,FE}$
EV	5550	$--$	362	362	362

Continued on next page

Table 5.7 – *Continued from previous page*

VV	2475	27150	225	798	1642
VE	1875	16875	741	4586	29102
EJ	10143	119301	277	1837	49196
JS	103707	138777	42454	340078	11146629
Totals	**123750**	**302103**	**42454**	**340078**	**11146629**
Number of points in Pareto front			**205**		
Optimum for f_1			**9.566 km/s km/s**		

Table 5.6 can also be used to infer approximate run time for the scenario considered, as the overall procedure is dominated by Lambert arcs and defects computations. Few microseconds are usually needed to compute a single Lambert arc [250] on standard laptop (i.e., 4 GHz single-core) on compiled code, while for the defects computation only few nanoseconds are necessary. Therefore, no more than few seconds are needed for the optimization even for the finest step-size cases (e.g., Case 1), making the whole procedure efficient for most interplanetary missions considered here. In conjunction with the Tisserand-informed planetary exploration, the whole procedure lasts from few seconds to couple of minutes for the cases considered.

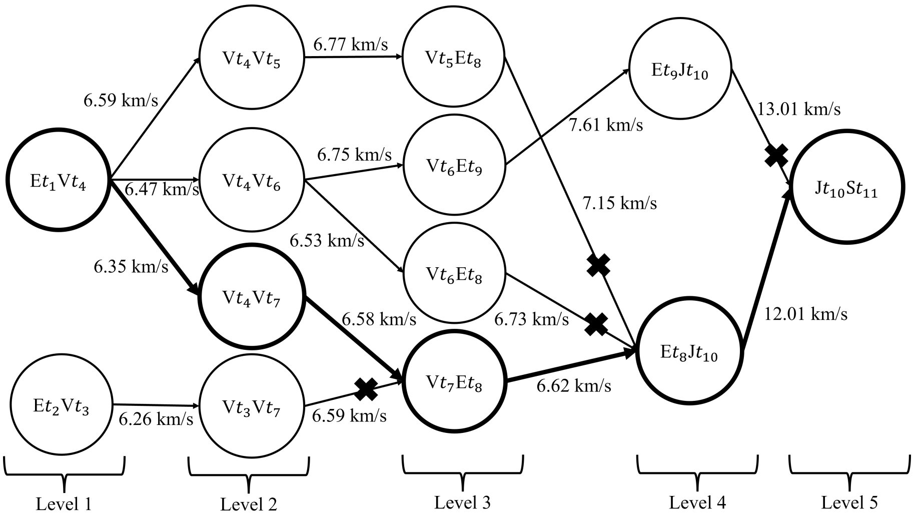

$\mathrm{E}t_1\mathrm{V}t_4$
$\mathrm{E}t_2\mathrm{V}t_3$
$\mathrm{V}t_4\mathrm{V}t_5$
$\mathrm{V}t_4\mathrm{V}t_6$
$\mathrm{V}t_4\mathrm{V}t_7$
$\mathrm{V}t_3\mathrm{V}t_7$
$\mathrm{V}t_5\mathrm{E}t_8$
$\mathrm{V}t_6\mathrm{E}t_9$
$\mathrm{V}t_6\mathrm{E}t_8$
$\mathrm{V}t_7\mathrm{E}t_8$
$\mathrm{E}t_9\mathrm{J}t_{10}$
$\mathrm{E}t_8\mathrm{J}t_{10}$
$\mathrm{J}t_{10}\mathrm{S}t_{11}$
6.59 km/s
6.47 km/s
6.35 km/s
6.26 km/s
6.77 km/s
6.75 km/s
6.53 km/s
6.58 km/s
6.59 km/s
7.61 km/s
7.15 km/s
6.73 km/s
6.62 km/s
13.01 km/s
12.01 km/s
Level 1
Level 2
Level 3
Level 4
Level 5

Figure 5.10: Example of EVVEJS tree exploration. Crossed paths are pruned by SODP application. Accumulated Δv up to the given tree level are also shown. The bold path is f_1-optimal.

The effect of Δv_{max} is also shown in Figure 5.11, representing the Pareto set degradation with respect to the Case considered and Δv_{max}. It can be seen from 5.11.a that increasing the defects limit at a value larger than 2 km/s does not produce significant variations of the Pareto front, while a too aggressive pruning might result in non-precise Pareto set representation. The leveraging ratio varies typically between 0.5 and 6.5 in most of the interplanetary missions considered here (see also section 5.5.3), so for example a maximum defect of 2 km/s implies a maximum DSM magnitude of about 0.310 km/s in the extreme leveraging case (note that 6.5 is still a high leveraging ratio and will be generally less than that). Thus, $\Delta v_{max} = 2$ km/s seems appropriate to truthfully represent Pareto front characteristics for the mission at hand.

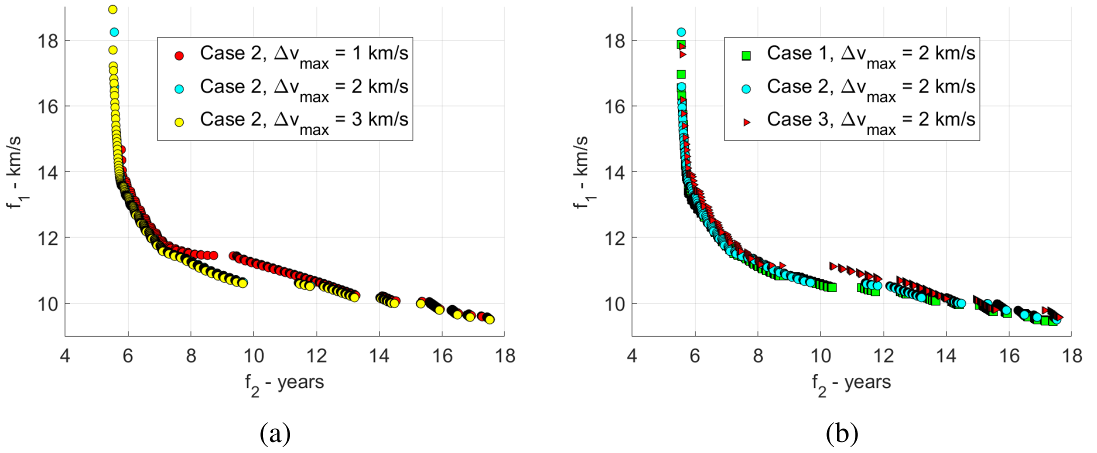


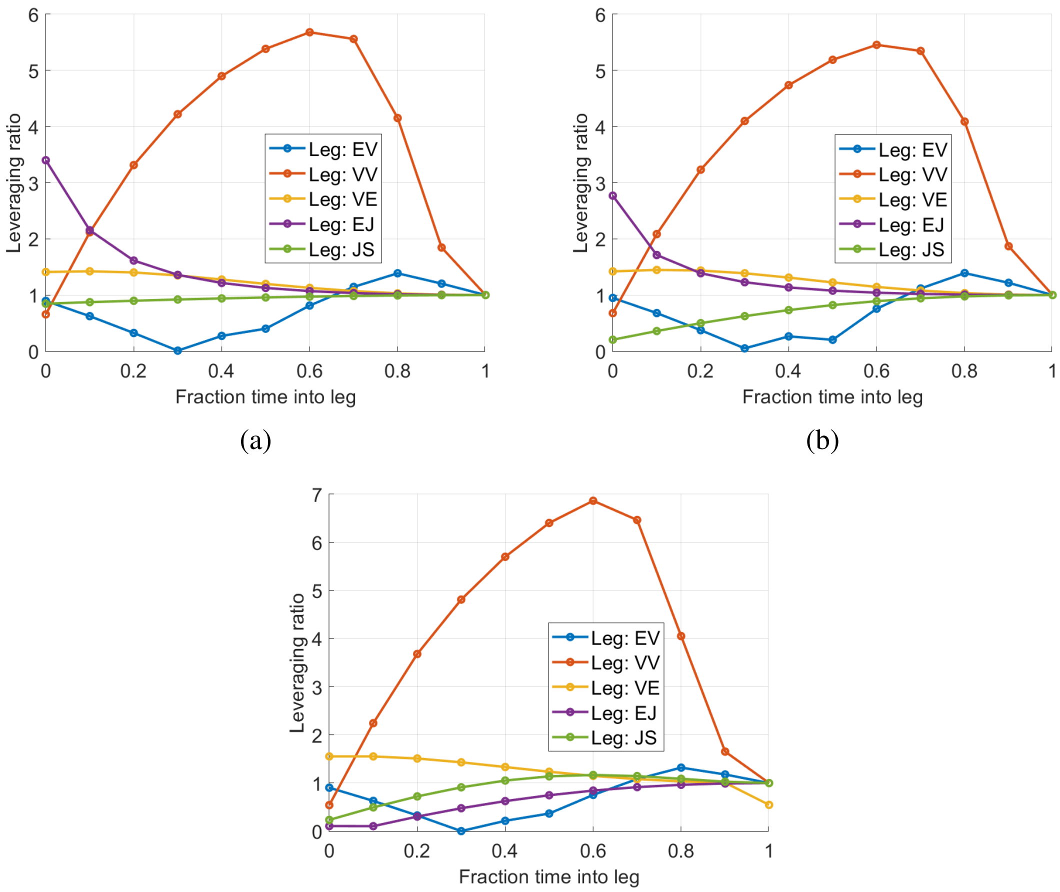
Figure 5.11: Pareto fronts for EVVEJS scenario varying with the Δv_{max} (a) and Case number (b).

Figure 5.11.b highlights the impact of the step size with respect to the Pareto representation. A fine search (Case 2) is usually preferable, as coarser step sizes can degrade the solutions quality relatively fast (Case 3). However, this choice is always a compromise between computational effort and solution quality, thus in this sense Case 2 seems the most balanced. The results also show the f_1 optimum solutions converging with increasing defect size for different interval considered, giving high confidence in the optimality for a given step size. To conclude, if defects are to be used as the only processing stage, i.e., with no further optimization in MGA-DSM, then small intervals (Case 2) are useful to use with MODP, providing accurate results (section 5.5.5) and are less consequential in terms of route numbers than other strategies. If another stage is envisaged, then a larger interval could also potentially be used (Case 3) with greater efficiency for the whole process. The issue with using large intervals in the first phase and then using a second phase, is to ensure that the large interval in the first phase does not result in selection of alternative local minima that do not then converge to the true minima in the second phase. In practice this does not seem happen with the interval ranges discussed.

5.5.3 Converting Defects into DSMs

A key feature of the pipeline described in this paper is the relationship between the infinity velocity defects and the DSMs obtained with the refinement step. Assessing the relationship between these two manoeuvre types is thus important to the understanding of the whole pipeline. The analytical approximation described in section 5.4 is employed here. This allows to obtain the leveraging effect obtainable from a DSM, i.e., the dependency of an infinity velocity defect on a preceding DSM. As it can be seen from both Table 5.3, 5.4 and 5.5, large defects occurring in the Venus-Venus leg of EVVEJS (both for Cassini, Cassini-2, and optimal solution) are replaced by DSMs, reducing from more than 1 km/s to approximately 400-600 m/s. The Δv occurring in all the other legs are virtually reduced to zero by means of the refinement process. In the fully optimized solution after the refinement, the defect at the start of the second leg, i.e., VV, has been removed by lowering the infinity velocity at the first Venus fly-by, with a DSM applied in the first leg. This happens in the case of both Cassini-2 and optimal solution, while nominal Cassini experiences higher velocity increment at the Earth departure. The lower initial infinity velocity at Venus causes a large defect at the second Venus encounter, but this is corrected by a DSM in the second leg where high efficiency of DSM is seen. The grid-based solution has higher cost, i.e., Δv prediction, than the refined solution, because the refinement redistributes the infinity velocity defects to maximally utilise the DSM leveraging.

This is noticed from the leveraging ratios computed on each leg of the EVVEJS transfer both for Cassini, Cassini-2, and optimal solution in Figure 5.12. The parameter represented is the infinity velocity defect corrected to magnitude of a DSM versus the time



(a)

(b)

(c)

Figure 5.12: Leveraging ratios per leg considered for Cassini (a), Cassini-2 (b) and optimal solution (c).

fraction elapsed into the segment. The plot illustrates that in the second segment, i.e., the Venus-Venus leg, the peak ratio is the highest in all the cases considered, and thus it implies that any infinity velocity defect can be efficiently corrected with a much smaller DSM, which is what happens in the refinement step. In the Earth-Venus leg, the ratio is generally less than 1 but exceeds 1 towards the end of the leg, reaching approximately 1.4. The efficiency in this segment is clearly much less than Venus-Venus segment. It is also interesting to notice that in the last Jupiter-Saturn leg for both Cassini and Cassini-2 scenarios (Figure 5.12.a and 5.12.b), the leveraging ratio remains below 1 for all the time elapsed in the segment, reaching its maximum value, i.e., 1, only at Saturn encounter. This suggests that no DSM are useful to leverage infinity velocity at Saturn, as also confirmed by refined results in Table 3 and Table 4. In the case of optimal solution (Figure

5.12.c), the peak of the leveraging ratio on the last leg is higher than 1 around the mid-region of the last Jupiter-Saturn leg. Therefore, it is more efficient to remove most of the infinity velocity at Saturn with large DSM between last two planets as again proved by the refinement process in Table 5.5.

5.5.4 Multi-Objective Optimization of Earth-Saturn Missions

In the second optimization scenario considered here, one assumes that the sequence to reach Saturn is not known but needs to be selected as part of the optimization process. Table 5.8 highlights the set up for this scenario. One should notice that the time bounds are larger than those considered in Table 5.2, and multiple revolutions around the Sun (up until 2 full revolutions) are admissible for innermost planets (i.e., if Venus, Earth, or Mars are involved in the transfer). The launch can happen anytime in 1997 to benchmark solutions with Cassini-like transfers and the maximum number of planets in the sequence is selected to be 6. As in the previous case, for the grid optimization, step size in start date and durations, st_s and st_d, respectively, are chosen to be $st_s = st_d = 3$ days for transfers involving Earth, Venus and Mars, while $st_d = 6$ days is suitable for transfers involving Jupiter and Saturn.

Table 5.8: Optimization scenario for Earth-Saturn mission for a launch in 1997.

Design variables	Values and bounds
Sequence	Unknown. Any planet can be chosen among: Venus, Earth, Mars, Jupiter, Saturn. Maximum number of planets: 6
Departure velocity magnitude	$v_{\infty,dep} \in [3,5]$ km/s
Maximum defect at each fly-by	$\Delta v \in [0,2]$ km/s
Launch window	$t_0 \in [-1095.5, -730.25]$ MJD2000
Number of revolutions about the Sun	$N_{rev} \leq 1$ if only V, E, M are in the leg; $N_{rev} = 0$ if J or S are in the leg
Transfer times between planets	$T \in [50,750]$ days If any leg has V, E, M $T \in [500,5000]$ days If any leg has J or S $T \in [400,2500]$ days If $N_{rev} \geq 1$

With such a scenario, a total of 11 sequences are identified, for which Pareto fronts are shown in Figure 5.13. Pareto fronts result from the specific mission scenario considered in Table 5.8, that considers less than one revolution when either Jupiter or Saturn are present in a leg. One expects further reduction in f_1 in the region of 24-25 years of transfer times if such hypothesis is removed. As it can be noticed, Cassini-like transfers following an EVVEJS sequence still represent the f_1 optimal solution for short mission durations, i.e., approximately 7 years of transfer time, while other sequences like EVEJS or EVEEJS become competitive for longer mission durations, i.e., more than 10 years. In general, Tisserand based exploration already informs the optimization process that having Jupiter as last planetary fly-by reduces the infinity velocity at Saturn, thus providing better transfers in terms of f_1 objective.

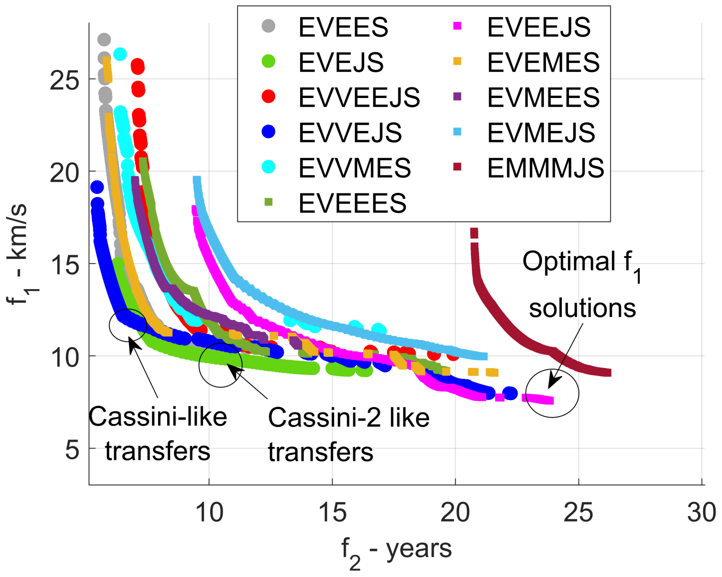


Figure 5.13: Pareto front of Earth-Saturn options.

One can see that in fact EVEJS seems to dominate EVVEJS for mission durations at about 10 years of transfer times. This mission option is characterized by large defect on the EJ segment as reported in Table 5.9, as a single Earth fly-by usually is not necessary to increase the spacecraft apoapsis to Jupiter orbit without the use of a large manoeuvre. In fact, a quick analysis of the leveraging ratio in Figure 5.14 shows that an optimal manoeuvre location in this segment should be at the very beginning of the leg, i.e., in the same position of the defect, to remove the velocity defect at the next Jupiter encounter. However, the defect at Jupiter is already small, thus the manoeuvre is likely not to decrease as much as in the EVVEJS case, where high efficiency is seen on the VV leg, as experienced in the refinement step shown in Table 5.3. Trajectories for EVEJS both before and after the refinement are shown in Figure 5.15, where one can see the manoeuvre at the very beginning of the EJ segment. One can appreciate that the trajectory structure is very similar to the EVVEJS as in section 5.5.1, and in fact the additional Venus fly-by before the Earth encounter is useful to increase the infinity velocity at the Earth, thus reducing the defect on the EJ leg.

Table 5.9: Results for the $\approx$ 10 years EVEJS solution in the given mission scenario.

Event	**Defects solution**	**Refined solution**
Earth departure	Oct. 22, 1997	Oct. 20, 1997
$v_{\infty,dep}$	3.73 km/s	3.99 km/s
Δv_1	$--$	$--$
Venus fly-by	Mar. 26, 1998	Mar. 24, 1998
Δv_2	0.425 km/s	$--$
Earth fly-by	Aug. 04, 1999	Aug. 02, 1999
Δv_3	1.63 km/s	1.44 km/s
Jupiter fly-by	Apr. 09, 2001	May 12, 2001
Δv_4	$--$	$--$ km/s
Saturn arrival	Sep. 09, 2007	Oct. 14, 2008
$v_{\infty,arr}$	4.18 km/s	4.13 km/s
f_1	9.96 km/s	9.56 km/s
f_2	9.88 years	10.9 years

In addition, it is possible to identify a new f_1 optimal solution for the Earth-Saturn mission scenario considered. This is the case of the EVEEJS sequence, exploiting more than 1 revolution in the EVEE legs, outperforming the missions identified in section 5.5.1, in terms of f_1 value, at the price of increased total transfer time f_2.

The trajectory is depicted in Figure 5.16 and corresponding values are shown in Table 5.10. Compared to the EVVEJS optimum from section 5.5.1, EVEEJS experiences a lower Δv both in the first EV and last JS leg, in terms of DSMs and infinity velocities at the planet. A general consideration is that the topology of the search space is correctly captured for any of the sequence identified, and the grid optimization alongside the dynamic programming approach already provides a very powerful tool to correct trade off analysis with very small computational effort (see section 5.5.2).

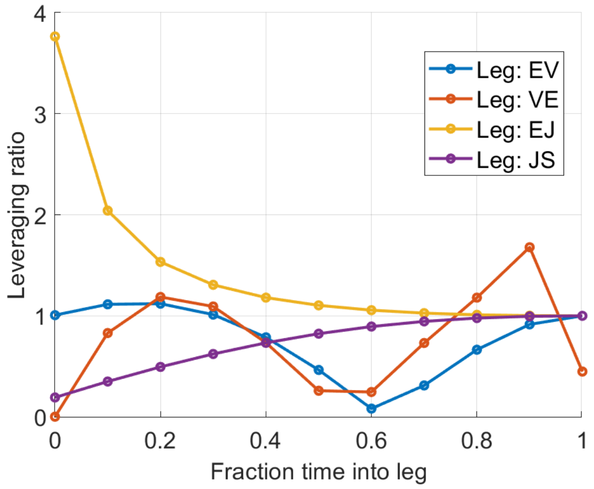
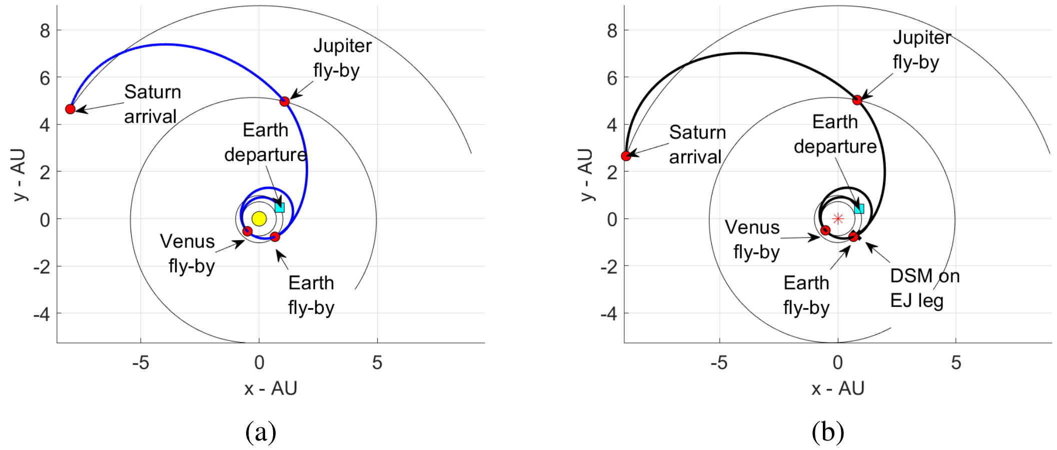


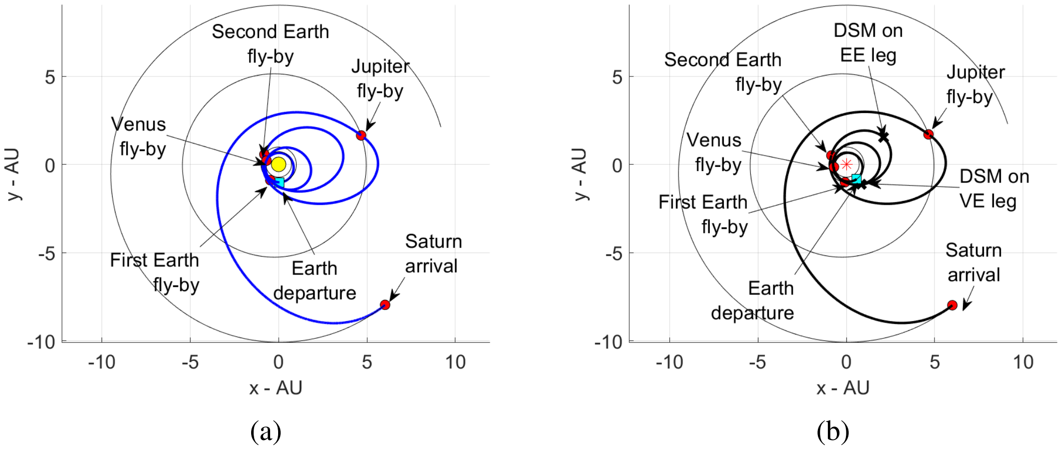
Figure 5.14: Leveraging ratios for EVEJS transfer for a transfer duration of ≈ 10 years.



(a)

(b)

Figure 5.15: EVEJS from grid (a) and refinement (b) for a transfer duration ≈ 10 years.



(a)

(b)

Figure 5.16: EVEEJS from grid (a) and refinement (b) for a transfer duration ≈ 25 years.

Continued on next page

Table 5.10 – *Continued from previous page*

Table 5.10: Results for optimal EVEEJS in the given mission scenario.

Event	Defects solution	Refined solution
Earth departure	Jun. 24, 1997	Jul. 24, 1997
$v_{\infty,dep}$	3.77 km/s	3.75 km/s
Δv_1	– –	– –
Venus fly-by	Sep. 27, 1998	Oct. 10, 1998
Δv_2	0.609 km/s	0.525 km/s
Earth fly-by	May 23, 2000	Jun. 16, 2000
Δv_3	0.0751 km/s	0.057 km/s
Earth fly-by	Feb. 14, 2007	Feb. 18, 2007
Δv_4	– –	– –
Jupiter fly-by	May 08, 2011	May 15, 2011
Δv_5	– –	– –
Saturn arrival	Apr. 23, 2021	Apr. 16, 2021
$v_{\infty,arr}$	3.09 km/s	3.08 km/s
f_1	7.54 km/s	7.41 km/s
f_2	23.8 years	23.7 years

This is useful in preliminary mission analysis when multiple mission options are generally required in short time with very little a priori knowledge of the structure of the final trajectory. In this way, all the possible feasible trajectory options are identified and can inform successive optimization in higher-fidelity models.

5.5.5 Comet Sample Return Missions

A mission scenario towards cometary objects among the Jupiter Family Comets (JFCs) is explored in this section. These objects are characterized by periapsis of approximately 1 AU and apoapsis close to the orbit of Jupiter, which makes them accessible for sample return and rendezvous opportunities. Such missions are of paramount importance for future Solar System exploration to understand the assembly process of cometary materials and their relationship with large-scale mixing in the ancient Solar System. As such, they have received attention from scientific community in the context of ESA Cosmic Vision 2050 for future missions [210].

The scenario considered here is reported in Table 5.11. The increased complexity of this mission scenario mainly relies on the high number of objects' encounters, the extended launch window and transfer times as well as the trajectory structure in terms of

Table 5.11: Optimization scenario for comet sample return mission for a launch in 2030-2040.

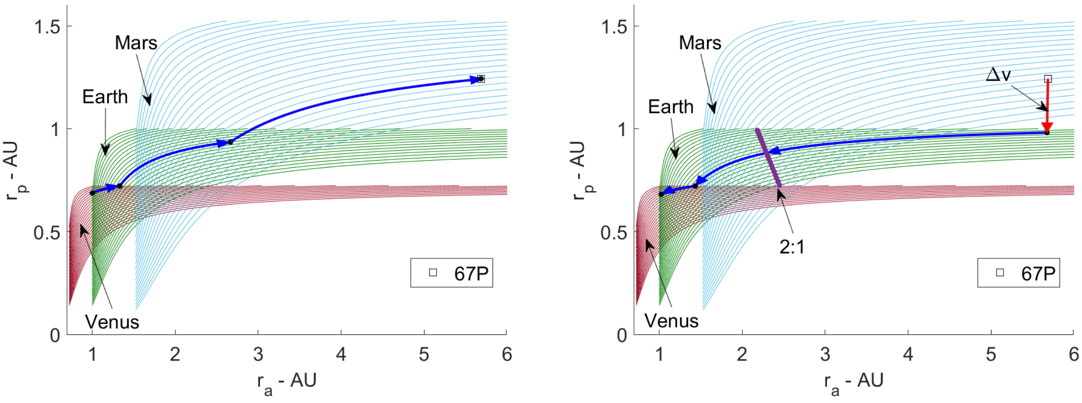
Design variables	**Values and bounds**
Target comet	67P/Churyumov-Gerasimenko
Sequence	Unknown. Any planet can be chosen among: Venus, Earth, Mars. Maximum number of objects: 10
Departure velocity magnitude	$v_{\infty,dep} \in [3,5]$ km/s
Maximum defect at each fly-by	$\Delta v \in [0,2]$ km/s
Launch window	$t_0 \in [10957.5, 14610]$ MJD2000
Number of revolutions about the Sun	$N_{rev} \leq 1$ if only V, E, M are in the leg; $N_{rev} = 0$ if 67P is in the leg
Maximum time on each transfer phase (either to go to the comet or to return to the Earth)	$t_p \in [0,8]$ years
Science phase time	$t_w \in [6,12]$ months
Transfer times between planets	$T \in [50,750]$ days If any leg has V, E, M $T \in [300,2500]$ days If any leg has 67P $T \in [400,2500]$ days If $N_{rev} \geq 1$

spacecraft revolutions about the Sun [4]. The target comet is assumed to be known, and is 67P/Churyumov-Gerasimenko, which is the same target of ESA's Rosetta mission [249]. One looks for trajectories in the 2030-2040 launch window. The constraints considered for this optimization problem are: (1) the maximum defects limit is set at 2 km/s for each leg considered, (2) the transfer time on each phase of the mission t_p (either to go to the comet or to return to the Earth) should not exceed 8 years, (3) the time t_w between cometary rendezvous and departure should be within 6 to 12 months to account for science phase operations. The objective is to explore the whole 10-years launch window to find suitable trajectory options to go to the comet and return within the constraints. Thus, a MODP-based approach is considered with the following objectives: (1) the first objective is to maximise the spread in launch dates, i.e., to look how many opportunities exist to go to 67P and return within the constraints in Table 11; (2) minimize the overall mission cost computed as follows:

[4] The maximum N_{rev} derive from pure mission analysis considerations: any additional revolution imply (more than) 1 year of transfer time, likely making the trajectory last too long.

$$f_1 = \begin{cases} v_{\infty,dep} + \sum_{i=1}^{n_{int}-1} \Delta v_i + \Delta v_{arr} + \Delta v_{dep} + (v_{\infty,arr} - 4\,km/s) & \text{if } v_{\infty,arr} > 4 \text{ km/s} \\ v_{\infty,dep} + \sum_{i=1}^{n_{int}-1} \Delta v_i + \Delta v_{arr} + \Delta v_{dep} & \text{otherwise} \end{cases} \tag{5.10}$$

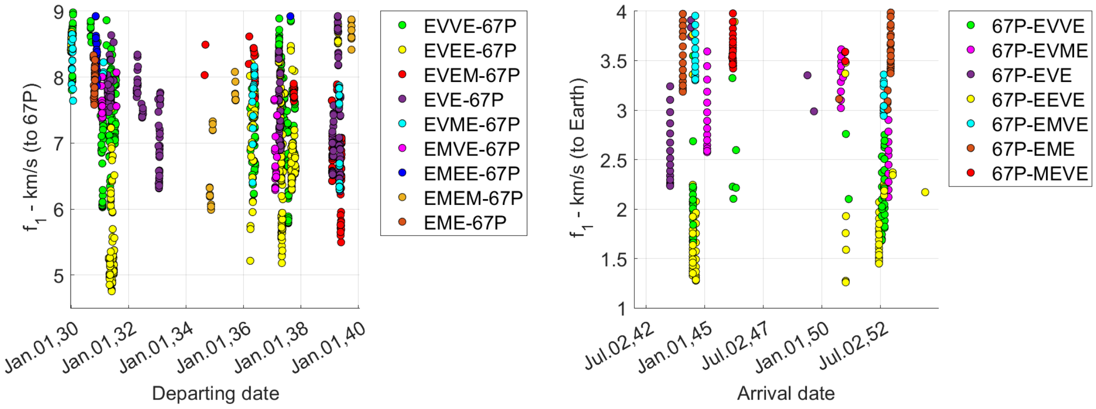
where $v_{\infty,dep}$ and $v_{\infty,arr}$ are the departing and arrival infinity velocities at the Earth, respectively; Δv_i are the manoeuvres on each leg of the transfer (either on the way to the comet or on the return phase to the Earth); Δv_{arr} and Δv_{dep} are the manoeuvres required for the rendezvous and the departure with the comet, respectively. The f_1 function accounts for both the phases of the mission, i.e., the transfer to the comet and the return phase to the Earth, and considers a free return to the Earth if the $v_{\infty,arr} \leq 4\,km/s$ (considered as a reasonable maximum threshold for a free-Δv re-entry in the Earth atmosphere [251,252]). To ensure correct spreading in the departing dates over the large launch window (10 years), all the paths that depart the Earth with different launch epochs and that are compliant with the constraints are retained at any selection steps of the MODP expansion.



(a) (b)

Figure 5.17: EVEM (a) and EEVE (b) sequences for a 67P sample return mission as from Tisserand exploration.

The MODP exploration informed by Tisserand-based criterion automatically allows to identify a very high number of trajectories (109) involving up to 6 fly-bys with Solar System planets for a total of up to 9 objects in the overall sequence. The overall launch window exploration remains efficient (see section 5.5.2) and only takes approximately 1.5 hours (4 GHz laptop).



(a) (b)

Figure 5.18: Cost of transfers towards 67P (a) and back to the Earth (b) with respect to launch and arrival date, respectively.

Figure 5.17 represents a possible strategy to go to the comet and to return to the Earth, involving an EVEM transfer towards 67/P (5.17.a) and an EEVE transfer on the way back, exploiting a 2:1 resonance on the EE leg of the mission (Figure 5.17.b). From Figure 5.17, one recognizes that having Mars as last planetary encounter (or first encountered planet on the return phase) could be useful to rendezvous/depart at 67P/Churyumov-Gerasimenko with virtually zero Δv, having a fly-by with the planet at about 12 km/s of relative velocity. On the other hand, transfers with Earth and Venus immediately before/after the cometary encounter imply a larger Δv and relatively high relative velocities at the planets. However, transfer scenarios only involving Earth and Venus seem to have lower overall Δv as the synchronicity between those two planets is more favorable than the one with Mars.

Figure 5.18.a illustrates the f_1 cost of reaching the comet with the sequences identified by the pipeline with respect to the launch date at the Earth, while Figure 5.18.b shows the cost of the return phase with respect to the arrival date at the Earth. The two phases are here separated for the sake of representation, and the f_1 cost refers to the single phase under consideration. One notices that the specular sequences like EVEE and EEVE provide the cheapest transfers towards the comet and back to the Earth, respectively. When

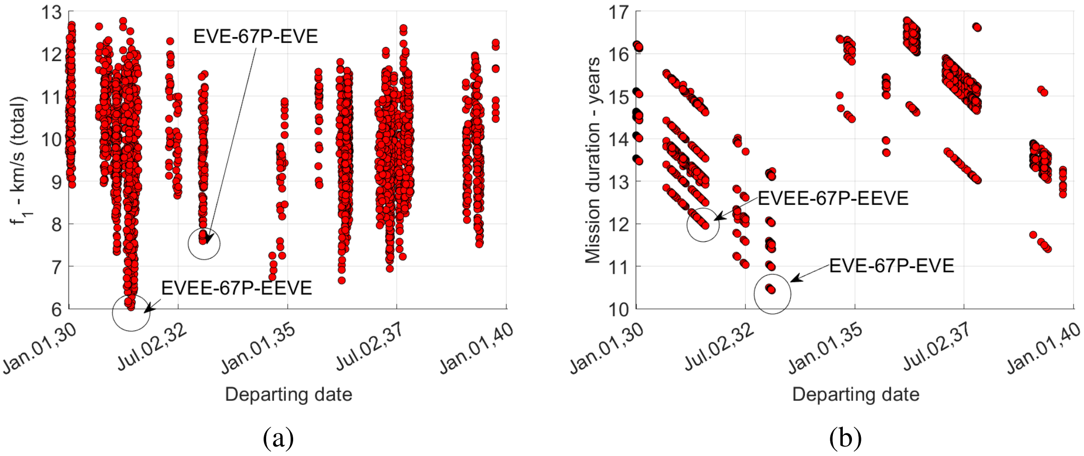


Figure 5.19: Total f_1 cost (a) and mission duration (b) for comet sample return options with 67P in the 2030-2040 launch window.

compared to other transfers, sequences having Mars as last/first planetary fly-by, such as the specular EVEM or MEVE, usually experience higher defects at Mars encounter due to rare phasing with Earth and Venus. This is also shown in 5.19.a, where where EVEE-67P-EEVE sequence appears as the optimal with respect to f_1 objective, exploiting 2:1 resonant transfer on each EE leg.

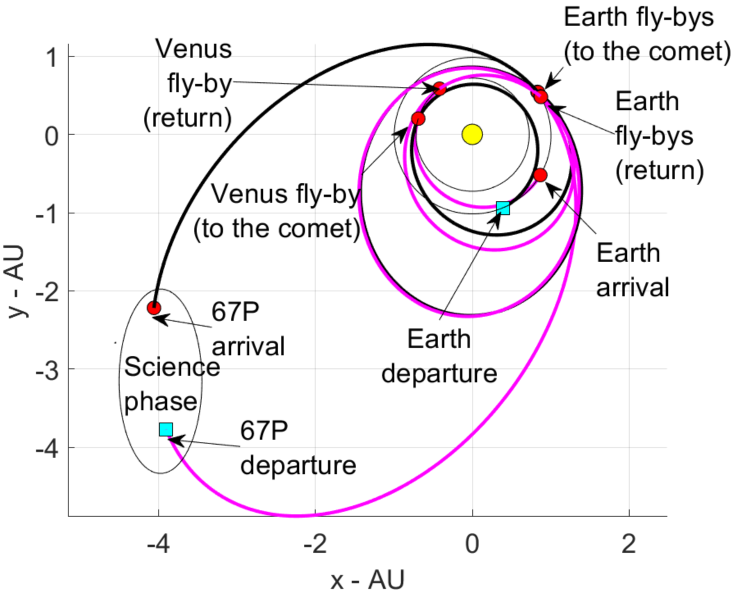


Figure 5.20: Comet sample return trajectory to 67P involving an EVEE trajectory to the comet (black path) and a EEVE transfer on the way back (magenta path).

This transfer is shown in Figure 5.20, and the corresponding events and values are reported in Table 5.12. No large manoeuvres are required neither in the defect approximation, nor in the refined solution, showing very good correspondence between defect solutions and refined ones, and the only contribution to the f_1 cost is given by the rendezvous with the comet and the departure on the return phase of the mission.

Table 5.12: Results for optimum solution for the given comet sample return mission scenario.

Event	**Defects solution**	**Refined solution**
Earth departure	Jul. 15, 2031	Jul. 05, 2031
$v_{\infty,dep}$	4.31 km/s	3.89 km/s
Δv_1	−−	−−
Venus fly-by	Dec. 17, 2031	Dec. 17, 2031
Δv_2	−−	−−
Earth fly-by	Oct. 26, 2032	Oct. 28, 2032
Δv_3	−−	−−
Earth fly-by	Oct. 26, 2034	Oct. 28, 2034
Δv_4	0.743 km/s	−−
Comet arrival	May 08, 2011	May 11, 2011
Δv_{arr}	1.33 km/s	1.24 km/s
Comet departure	Mar. 16, 2037	Mar. 16, 2037
Δv_{dep}	1.14 km/s	1.14 km/s
Δv_5	−−	−−
Earth fly-by	Oct. 22, 2040	Oct. 22, 2040
Δv_6	−−	−−
Earth fly-by	Oct. 22, 2042	Oct. 22, 2042
Δv_7	0.0874 km/s	−−
Venus fly-by	Mar. 13, 2044	Mar. 07, 2044
Δv_8	0.0262 km/s	−−
Earth arrival	Aug. 21, 2044	Aug. 11, 2044
$v_{\infty,arr}$	4.02 km/s	3.78 km/s
f_1	6.48 km/s	6.27 km/s
f_2	13.1 years	13.1 years

From Figure 5.19, another interesting option is represented by EVE-67P-EVE sequence, which experiences short transfer times and good f_1 costs, due to small number of planetary encounters, but at the price of rare repetition over the mission range considered. Figure 5.19.b also shows that almost every year an opportunity to rendezvous with the comet and return to the Earth exists within relatively short mission durations, i.e., less than 15 years, within the Δv constraints. The procedure has proven again to be efficient in exploring a mission scenario of complex configuration, demonstrating the ability of identifying multiple mission options for practical preliminary analysis of future missions.

5.6 Planning High-Inclination Tours

Tisserand-based exploration as from Chapter 4 allows to identify the need for resonant orbits on legs that visit the same planet consecutively. These can occur when the time of flight is such that the spacecraft encounters the same point for two encounters. Such transfers are thus characterized by a ratio N:M of integers between the planet and spacecraft orbit periods, where N is the number of planet revolutions about the Sun and M is the number of spacecraft revolutions.

The details of resonant orbits can be obtained analytically via derivation of post fly-by relative velocity vector characteristics, and thus solution of Lambert's problem is not required. A brief description of the process to obtain resonant transfers is reported in section 5.6.1, while a possible application for exploiting high-inclination orbits is presented in section 5.6.2.

5.6.1 Generating Resonant Orbits

The resonant ratio N:M implies a ratio between the planet and spacecraft orbital periods, namely T_{pl} and T_{res} respectively, that is (from Chapter 4):

$$T_{res} = \frac{N}{M} T_{pl} \tag{5.11}$$

It is thus possible to compute the semi-major axis of the resonant orbit as:

$$a_{res} = \left(\mu \left(\frac{T_{sc}}{2\pi} \right)^2 \right)^{1/3} \tag{5.12}$$

The spacecraft velocity after the first fly-by with the resonant planet is thus:

$$v^+ = \sqrt{\mu \left(\frac{2}{|\vec{r}_{pl}|} - \frac{1}{a_{res}} \right)} \tag{5.13}$$

Where $\vec{r}_{pl}$ is the planet position at the encounter. Recalling the velocity triangle of Figure 5.21, one has that the angle $\alpha = \pi - \theta$ on which:

$$\theta = \arccos\left(\frac{v_\infty^{-2} + |\vec{v}_{pl}|^2 - v^{+2}}{2v_\infty|\vec{v}_{pl}|}\right) \tag{5.14}$$

where v_∞^{-2} and $\vec{v}_{pl}$ are the infinity velocity magnitude and planet velocity vector at the planetary encounter, respectively. Similarly to what already seen in Chapter 4, an N:M resonant orbit is achievable for a given v_∞^- only if:

$$\left|\frac{v_\infty^{-2} + |\vec{v}_{pl}|^2 - v^{+2}}{2v_\infty|\vec{v}_{pl}|}\right| \leq 1 \tag{5.15}$$

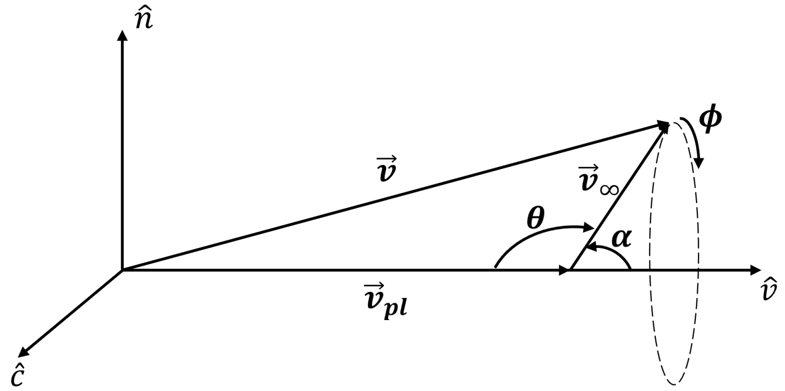


Figure 5.21: Possible spacecraft orientations after the first fly-by.

In a local reference frame (see Figure 5.21), defined as [153]:

$$\begin{cases} \hat{v} &= \frac{\vec{v}_{pl}}{|\vec{v}_{pl}|} \\ \hat{n} &= \frac{\vec{r}_{pl}\times\vec{v}_{pl}}{|\vec{r}_{pl}\times\vec{v}_{pl}|} \\ \hat{c} &= \hat{v}\times\hat{n} \end{cases} \tag{5.16}$$

one has that the infinity velocity vector after the first fly-by with the resonant planet, i.e., $\vec{v}_\infty^+$ is found as:

$$\vec{v}_\infty^+ = v_\infty^- \begin{bmatrix} \cos\alpha \\ \sin\alpha\cos\phi \\ \sin\alpha\sin\phi \end{bmatrix} \tag{5.17}$$

One should then convert $\vec{v}_\infty^+$ in the inertial reference frame by:

$$\vec{v}_\infty^+ \leftarrow T\vec{v}_\infty^+ \tag{5.18}$$

where T is the transformation matrix made by vectors $[\hat{v},\hat{n},\hat{c}]$. The rotation of $\vec{v}_\infty^-$ to $\vec{v}_\infty^+$ should again be compliant with the maximum deflection admissible at the given fly-by. Finally, one can reconstruct relevant information, such as the spacecraft velocity after the first fly-by with the resonant planet, $\vec{v}^+ = \vec{v}_{pl} - \vec{v}_\infty^+$, that, alongside the spacecraft position $\vec{r}^+ = \vec{r}_{pl}$, fully define the resonant orbit. However, one should notice that such orbit is function of the angle ϕ, that defines spacecraft orbits of different inclinations. Therefore, one could either select the angle that minimizes the inclination change between the pre- and post-fly-by orbit or select a range of inclined resonant orbits to be kept for further consideration.

5.6.2 High-Inclination Tours

If the aim is to maximise the heliocentric inclination, the analysis presented in Chapter 4 already showed the effectiveness of using Venus and Earth as possible planets to fly-by to gradually increase the orbital inclination. Specifically, the Earth is particularly effective to change the inclination compared to Venus since it is farther from the Sun, permitting to achieve higher inclinations with similar infinity velocities with respect to Venus.

The steps to construct high-inclination tours would be to identify from Tisserand graph exploration (Chapter 4) the infinity velocities that allow to reach the desired inclination with the Earth (say, for example, that the target is to have orbits with inclination higher

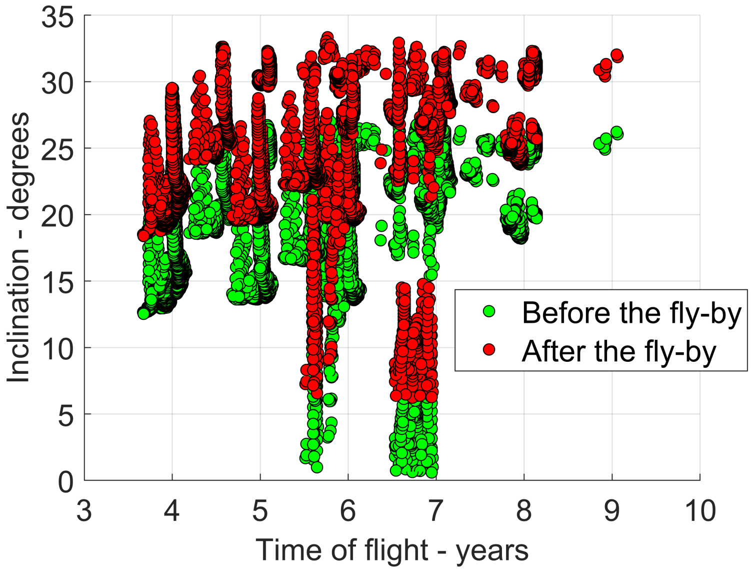


Figure 5.22: Inclination before and after the last Earth fly-by of EVEEEE with 1:1, 1:1 and 2:3 resonant orbits between successive Earth fly-bys.

than 30 degrees). In this case, the infinity velocity at the Earth should be at least 15 km/s. To achieve such speeds, Tisserand graph exploration allows to identify sequences like EVE or EVEVE, followed by sequences of 1:1 and 2:3 resonant orbits to maximize the inclination.

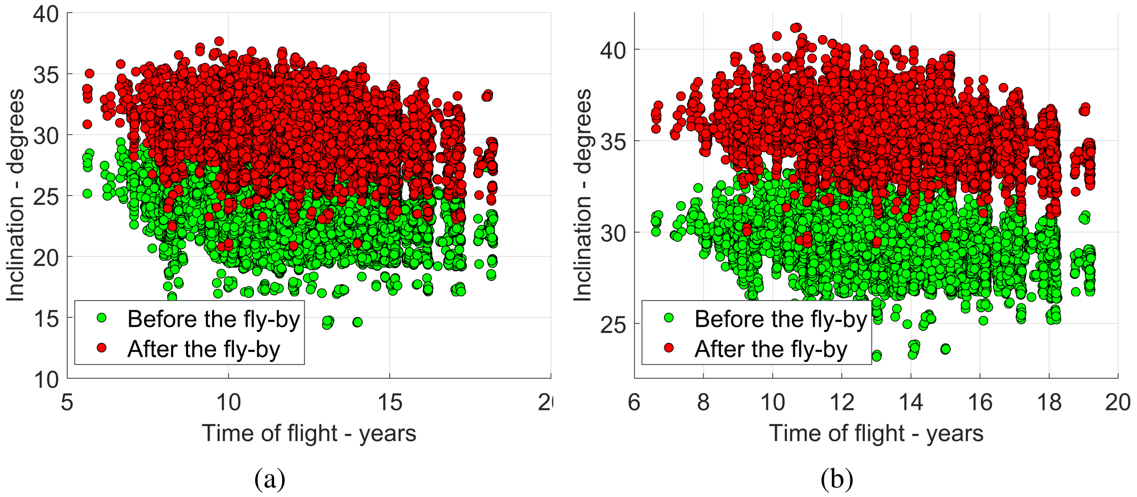


Figure 5.23: Inclination before and after the last Earth fly-by of (a) EVEVEEEE with 1:1, 1:1, and 2:3 resonant orbits on the successive EE fly-bys, and (b) EVEVEEEEE with 1:1, 1:1, 1:1 and 2:3 resonant orbits on EE legs.

Successive resonant orbits are constructed by means of the steps presented in section 5.6

Figure 5.22 shows the achievable inclinations for the option EVEEEE with 1:1, 1:1 and 2:3 resonances before and after the last fly-by with the Earth. The maximum inclination achievable is approximately 33.34 degrees in about 5.765 years.

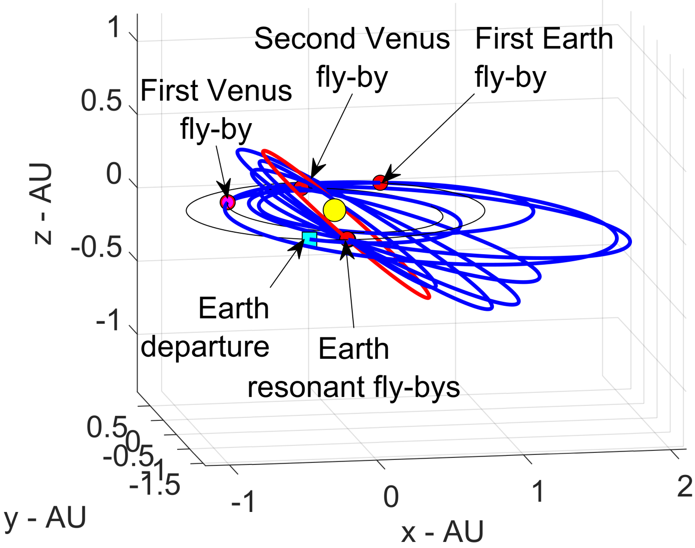


Figure 5.24: EVEVEEEEE sequence with 1:1, 1:1, 1:1 and 2:3 resonant orbits on successive EE legs. The red orbit is achieved after the last fly-by with the Earth.

For the option EVEVE, the inclination achievable is higher thanks to the higher infinity velocity at the last Earth encounter reached thanks to the Venus fly-by. One could use successive Earth fly-bys in 1:1, 1:1 and 2:3, as in Figure 5.23.a to achieve higher inclinations (the infinity velocity at the Earth is higher, i.e., 17 km/s), or use 1:1, 1:1, 1:1 and 2:3 to achieve a maximum of 37.65 degrees in just under 10 years, or above 35 degrees in less time, i.e., 5.695 years. By adding another 1:1 resonant fly-by before the 2:3 one, Figure 5.23.b shows that the maximum inclination becomes 41.19 degrees, reached in 10.16 years, or one can get to approximately 37 years in 6.695 years. Figure 5.24 illustrates the EVEVEEEEE trajectory that achieves 41.19 degrees of inclination.

5.7 Conclusions

This Chapter presented a robust and efficient approach to the multi-objective optimization of complex MGA transfers. This relies on the specific process that considers: (1) approximated Δv manoeuvres, i.e., velocity defects at planetary encounters, allowing efficient exploration of the search space in terms of launch window and transfer times; (2) efficient exploration of the transcribed search space by means of a novel multi-objective dynamic programming (MODP) approach applied to interplanetary missions.

Specifically, robustness is ensured by the evaluation of the relationship between the different manoeuvre model types. Efficiency is achieved by exploiting the sub-optimal structure of the MGA path planning step. In this way, MODP is used as a method to explore the transcribed search space guaranteeing the global Pareto optimality of competing mission objectives.

The ability of the proposed approach in identifying globally optimal paths was tested against mission towards Saturn, similar to Cassini mission, to prove the effectiveness in representing wide Pareto fronts of complex configuration for well-known trajectories. Indeed, in such cases the method allowed for finding the Δv-global optimum without any need of a priori knowledge of the solution (e.g., on the gravity-assist sequence, the departing date, transfer times and DSMs) in an efficient way, from wider search spaces compared to existing literature, as well as new transfer scenarios that populate very wide Pareto fronts.

Numerical results showed that the proposed pipeline is also suitable for exploring novel scenarios such as sample return missions towards comets, where the complexity mainly lies in (1) the extended launch window considered, and (2) the structure of the trajectories themselves, requiring up to six fly-bys for the overall mission.

Finally, the extension of the presented tool to planning of high-inclination resonant tours is provided, showing the scalability of such approach to many different mission scenarios.

Chapter 6

Modified Dynamic Programming for MAB Exploration

The present Chapter focuses on the space trajectory design of asteroids' tour mission, where the term tour simply indicates a mission that aims to visit not one but several celestial objects by means of one single spacecraft. As identified in Chapter 2, these problems have been tackled as global optimization problems, under the formulation of mixed-integer global optimization problems. However, beyond the aim of finding the global optimum, mission designers are usually interested in providing a wide range of mission design options reflecting the multi-modality of the problem at hand. In this sense, a CSP-like formulation is also relevant.

It should be highlighted that this Chapter is ideal to support mission analysis studies like CASTAway [9, 10], MANTIS [205] or Lucy [11], and its content is inspired by the challenges posed by such missions. For such missions, no pre-defined targets are to be visited, but rather a feasibility assessment is needed of the MAB exploration with close passages with asteroids within the envelope of practical mission boundaries (e.g., ESA M-class boundaries).

The key ideas behind the present Chapter can be then summarized with the following points. The subdivision of the Chapter is also provided:

- Firstly, the models used in this Chapter are recalled in section 6.1.
- One then wants to construct sequence of asteroids with the help of planetary fly-bys to increasing the chances of visiting multiple asteroids. This is explained in section 6.2.
- The problem of finding asteroids and planets sequences is transcribed into a TSP variant, i.e., a pure combinatorial problem, on which the asteroids and planets can be visited only at fixed epochs. To do so, an approach based upon the Minimum Orbital Interception Distance (MOID) is described in section 6.3. The structure of such space, that is analogous to the one presented in Chapter 5, is also summarized.
- On this transcribed space, one wants to find all the possible sequences (combinations) of asteroids and planets (or as many as possible) that are compliant with given mission constraints. In this sense, one solves both the CSP-like problem (multiple sequences of asteroids to be found) and the global optimization problem (the optimal sequence with respect to a given objective is searched to asses the minimum spacecraft performances). For this purpose, one might want to consider exhaustive tree-graph explorations with DF or BF. However, retaining all the possible sequences is usually unfeasible due to the enormous tree size (that depends upon the number of asteroids in the sequence). One might then consider applying BS as a standard way to find numerous solutions that are compliant with the constraints, but with no guaranty on the optimality. This is discussed in section 6.4.
- Therefore, the proposed approach is to slightly modify the paradigm of dynamic programming as explained in Chapter 5 to retain not only the globally optimal path on the transcribed problem, but also multiple solutions with less computational effort compared to standard BS-like procedures. In this sense, the CSP-like approach as presented in Chapter 2 is set up such that trajectory optimization is still carried out in a modified dynamic programming fashion, thereby retaining more individuals

within the trajectory optimization population. This is explained in section 6.5.

- Section 6.6 presents numerical results that demonstrate the efficiency and effectiveness of applying the modified dynamic programming approach to the problem at hand. This shows novel mission scenarios within realistic set of mission boundary conditions using Earth, Venus, and Mars gravity assist for future missions to explore the main asteroid belt. Sequences of up to 18 asteroids are found and wide launch windows are explored (i.e., 10 years).

- Section 6.7 summarizes main results and achievements.

6.1 Modelling MGA Trajectories with Multiple Asteroids

By recalling the formulation employed in previous Chapters 3 and 5, typical multi-target mission analysis parameters (like Δv or mission lifetime) crucially depend on variables that can assume both integer and continuous-varying values, i.e., X and y, respectively. In this Chapter, integer vector set X encodes the targets to be visited, i.e., asteroids and planets, while the continuous-varying of vector y encodes visiting epochs and other continuous variables which describe spacecraft manoeuvres, such as planetary gravity assist or DSMs. The mathematical model employed is again the MGA-DSM one (see Chapter 3) on which the fly-by with planets are modelled as impulsive manoeuvres that are governed by the quantities (r_p, γ), defining the minimum distance of the fly-by hyperbola and the orientation of the fly-by plane. Regarding the fly-by with asteroids, these are assumed to be massless points in this preliminary analysis (i.e., no sphere of influence). The trajectory between two consecutive asteroids then results from a Lambert arc between the asteroids positions (known from ephemerides) and a Δv manoeuvre is assumed to link incoming and outgoing arcs at a given asteroid encounter (similarly to the defects computation in Chapter 5).

Figure 6.1 represents an example trajectory as proposed for CASTAway mission for ESA

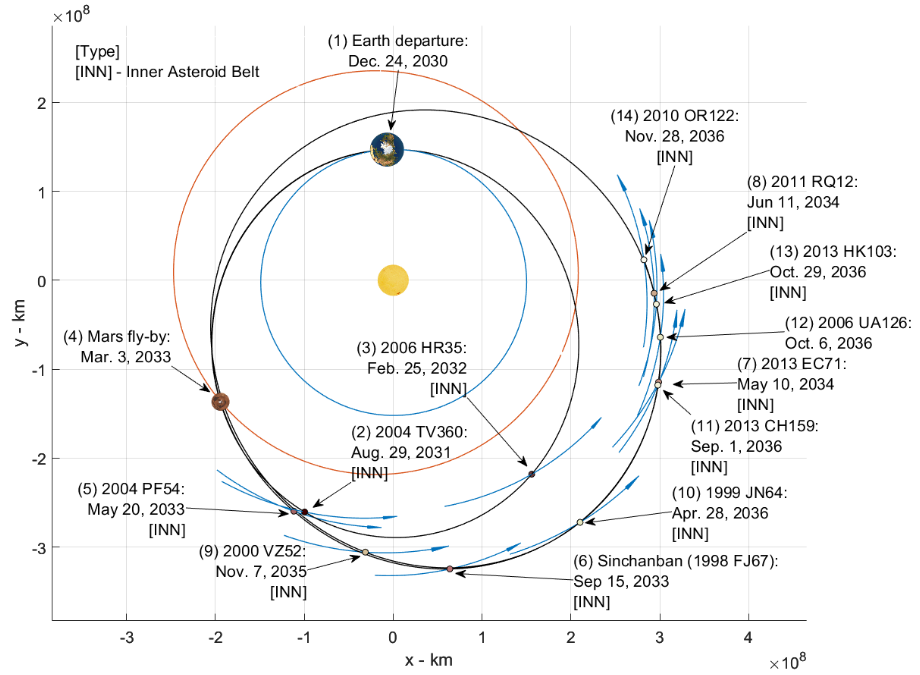


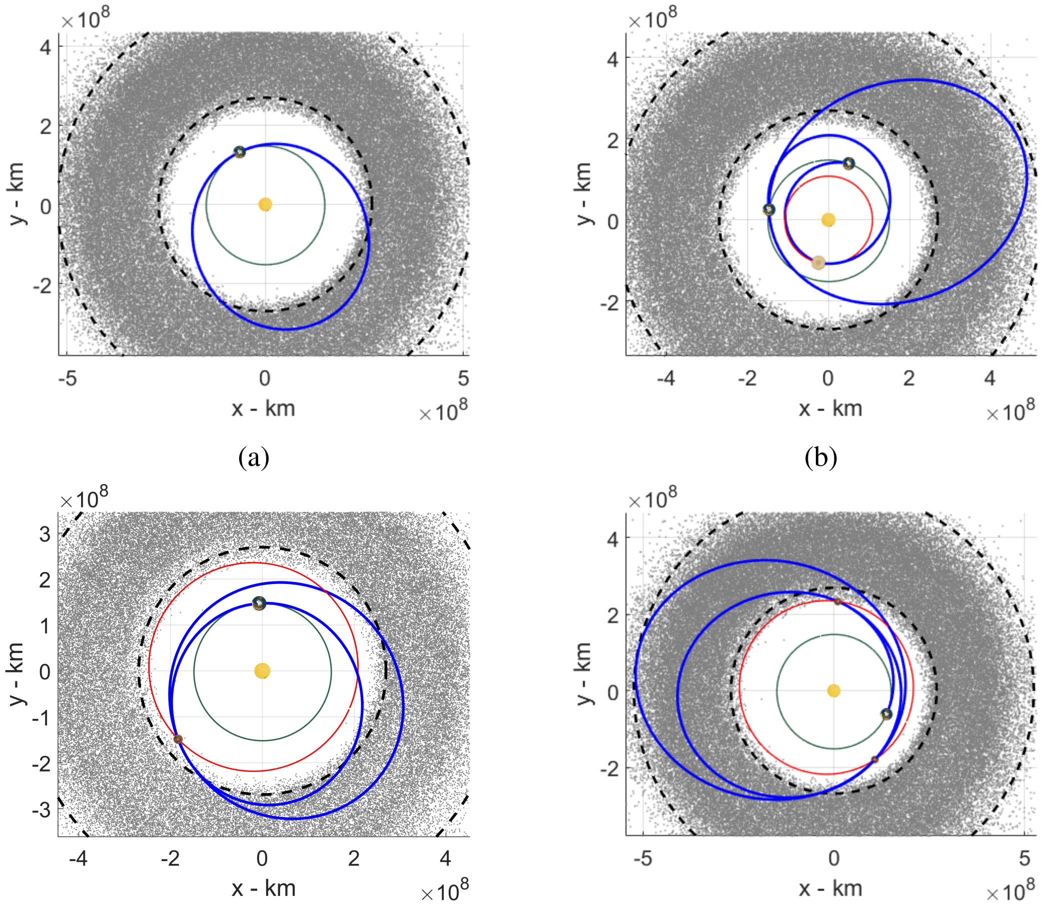
Figure 6.1: Example of MAB tour trajectory for CASTAway design envelop. Arrowed lines represent asteroids paths. Sun and planets textures are not in scale.

M5 call, which follows an Earth-Mars swing-by to increase the spacecraft orbital energy and encountering a total of 12 asteroids. In this case, vector X encodes a total of $n_{int} = 14$ visited objects (i.e., the Earth, that is the departing planet, 12 asteroids and Mars). Vector y encodes 17 variables, that are the departing date t_0, the transfer time between the objects $T_i, \forall i = 1, ..., n_{int} - 1$ and three variables for the Mars encounter, i.e., the fraction of time before the encounter at which a DSM is performed (η), the minimum distance to the planet during the fly-by (r_p) and the orientation of the fly-by plane (γ).

6.2 Planetary Gravity Assists Baseline Path

The (X, y) vectors contain the information necessary to describe a multiple planetary gravity assist trajectory with asteroids encounters. The few variables (both integer X and continuous-varying y) defining the planetary encounters will disproportionally contribute to the chances of the spacecraft to encounter main-belt asteroids at a low Δv expenditures. This is mostly due to the fact that a planetary encounter will greatly affect the overall energy and orientation of the spacecraft trajectory, hence affecting its capability to reach the MAB and therefore exploring it.

Ultimately, different MGA sequences allow the probe to enter the MAB with different trajectories and spend more or less time within the bounds of the MAB. Figure 6.2 shows three specific examples of planetary encounters to access the MAB. It showcases a simple Earth direct escape with no planetary encounters, as well as an Earth-Venus-Earth, an Earth-Mars and an Earth-Mars-Mars. The dotted lines represent the MAB limits, and so it can be observed that each transfer opportunity features different paths within it.



(a)

(b)

(c)

(d)

Figure 6.2: Four MGA trajectory options with E alone (a), EVE (b), EM (c), EMM (d). Grey points represent asteroids in the dataset and dotted lines are the limits of the MAB. Sun and planet textures are not in scale.

In the reminder of the Chapter, the Earth-Mars sequence as from Figure 6.2.c is considered (up to section 6.6.1), since as shown in Figure 6.2.c and analysed by Gallego [253] it enables a decent reach within the MAB at relatively low Earth $v_{\infty,dep}$, especially if short missions are considered (e.g., ≤ 5 years). The following sections describe the process to explore all asteroids tours possible for one single Earth departure in an Earth-Mars MGA sequence, and the details of the trajectory are reported in Table 6.1, where Δt_{MAB} is the time spent in the MAB region.

Table 6.1: Details for the Earth-Mars reference trajectory for M5 call.

Event	**Value**
Earth departure	Dec. 24, 2030
Departing v_∞	4.23 km/s
Mars fly-by	Mar. 3, 2033
Δt_{MAB}	1250 days

6.3 Transcription of the Problem

This section deals with asteroids tour problem transcription. It first defines the asteroids dataset (section 6.3.1), and then for any MGA path one selects potential asteroids encounter dates based upon MOID information (section 6.3.2). The accuracy of such transcription is then discussed in section 6.3.3.

6.3.1 Pre-pruning

Once a general baseline transfer, i.e., with planetary sequence and epochs for each planetary encounter, is identified, the search for asteroid belt multiple fly-by options can be initiated. As of September 2022, slightly over 1 million asteroids in the MAB are known, according to the Jet Propulsion Laboratory Small-Body Database [1]. Exploring all the potential, for example, 10-asteroids long sequences within the known population would require computing trajectories for more than 10^{53} different sequences. Computing this number of trajectories would take many orders of magnitude more than the age of the universe and, therefore, there is an obvious need to prune out the overall catalogue and

[1] `https://ssd.jpl.nasa.gov/tools/sbdb_query.html`, last accessed September 2022

target only a sensible set of interesting MAB objects. Here, a pruned database of $\sim$ 102,000 main belt asteroids is used to search for main-belt asteroid sequences [9, 10, 61]. This database provides a pre-filtered population of MAB objects: all asteroids larger than 10 km in diameter are retained in the database, while smaller objects are pruned out maintaining a representative diversity of asteroids in size and orbital distribution. The orbital elements and other physical information of the asteroids in the database is downloaded from Jet Propulsion Laboratory Small-Body Database. It should be noted that an asteroid set of $\sim$ 102,000 objects is already a much larger set than any GTOC-related asteroid set [254], and the addition of one or multiple gravity assists with planets as in Figure 6.2 adds extra complexity to the problem.

6.3.2 Asteroids Tour Transcribed Problem

The transcription process can now be initiated. Recall that two different sub-problems need to be solved to design an asteroids tour. Firstly, the right sequence of asteroids (i.e., completing the integer vector variable X), among the $\sim$102,000 targets, need to be appropriately chosen, which requires solving a discrete combinatorial problem. However, the goodness (Δv and/or TOF) of a given asteroid tour can only be assessed after identifying the actual dates for each asteroid encounter (i.e., continuous-varying vector variable y). It should be noted that the Δv cost of an asteroid tour will be highly sensitive to the actual dates of the asteroid encounters. A priori, the possible dates for each asteroid fly-by (t_{fb}) could be any date that satisfies $y_1 < t_{fb} < y_1 + TOF$ (TOF being the overall mission time of flight). The fact that both of these two sub-problems are tightly associated defines the crux of the asteroid tour problem and differentiates it from other classic combinatorial problems, such as the TSP.

Given that the asteroid tour sequence is refined over one specific baseline trajectory, as identified in section 6.2, one may consider that only asteroids that are close to the spacecraft at any one point during the trajectory can be feasibly encountered. An analytical algorithm identifying the minimum orbital intersection distance (MOID), combining

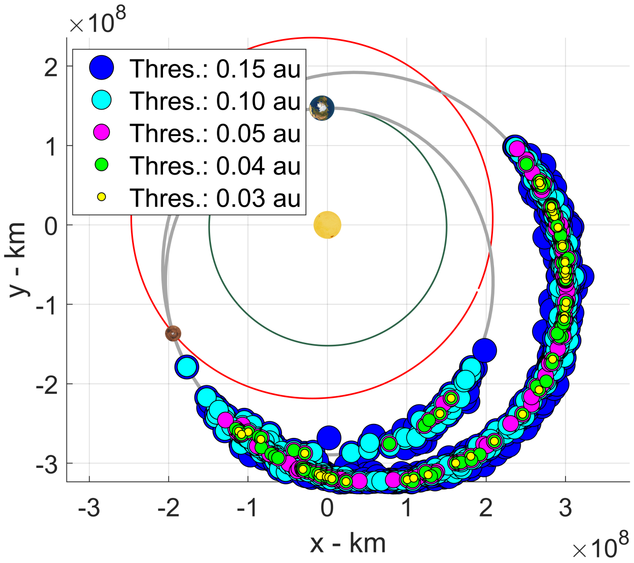


Figure 6.3: Reference trajectory and close asteroids with respect to the distance threshold. Sun and planets textures are not in scale.

both [255] and [256] algorithms, is instead implemented to identify asteroids within a MOID range, as well as their true anomaly at the epoch of the MOID point crossing. The distance between the spacecraft and the asteroid at the epoch at which the asteroid crosses its MOID point is here used as a good estimate of the asteroid minimum distance (note multiple asteroid epochs may exist within the spacecraft TOF).

Algorithm 4 summarizes the main steps of the transcription process. For any given asteroid in the catalogue, the algorithm computes the MOID i.e., the minimum distance between the asteroid and the spacecraft reference orbit on the leg. Then, it checks the position of the asteroid and the spacecraft at the asteroid's MOID point crossing epoch and computes the actual distance. If this is less than a pre-defined threshold d_{thr}, then the given asteroid is considered as a candidate object to be potentially reached. Algorithm 4 is thus used to compute sub-set of asteroids whose orbits intersect the spacecraft trajectory close enough to be suitable targets, with respect to the distance threshold.

Figure 6.3 shows the EM reference trajectory as from section 6.2 and close-by asteroids at their MOID point epochs with varying distance threshold values. Figure 6.3 shows the set of asteroids within the distance thresholds of 0.03 AU, 0.04 AU, 0.05 AU, 0.10 AU

Algorithm 4 Pseudo-code for computing potentially reachable asteroids and their MOID epochs with respect to a reference trajectory

```
Load reference trajectory (section 6.2), the population of MAB objects and $d_{thr}$:
for each leg on the GA reference trajectory do
    for each asteroid in the dataset do
        Compute the MOID between the leg and the asteroid orbit
        Compute spacecraft-asteroid distance at asteroid's MOID-point epoch
        if the distance is less than threshold $d_{thr}$ then
            Save the asteroid and its MOID-point crossing epoch
        end if
    end for
end for
```

and 0.15 AU. One can thus prune out all asteroids that do not satisfy a given distance threshold d_{thr}, which for the five thresholds in Figure 6.3 will leave an asteroid dataset containing 49, 98, 158, 562 and 1026 asteroids, respectively.

Note that each asteroid in the set also has an epoch associated to it, which identifies the epoch at which the asteroid crosses its MOID point. This uniquely identifies each asteroid with the time of its possible fly-by. Such an approximation allows thus tackling the path planning combinatorial problem on its own. The continuous optimization with the actual dates for each asteroid encounter may then be refined in a later stage (see section 6.3.3).

6.3.3 Refinement

By applying the transcription process, the problem is decomposed into two consecutive sub-problems: firstly, the discrete combinatorial problem (CP), which aims at identifying promising sequences of asteroids to complete a tour, and secondly, the refinement problem (RP) aiming at optimizing the continuous design variables given a fixed asteroid sequence. Hence, the RP refines all or a given promising sub-set of solutions from CP, which provide a fixed sequence of objects to visit. It is important to note that the CP solutions are not approximations of the complete problem, but full trajectories. The visiting epochs and planetary fly-by parameters, variables encoded in the vector y, identified in the CP process only need to be refined in case a lower Δv solution exist in the same neighbourhood. For this process to be efficient, CP solutions must be close enough to the real minimum solution for the RP.

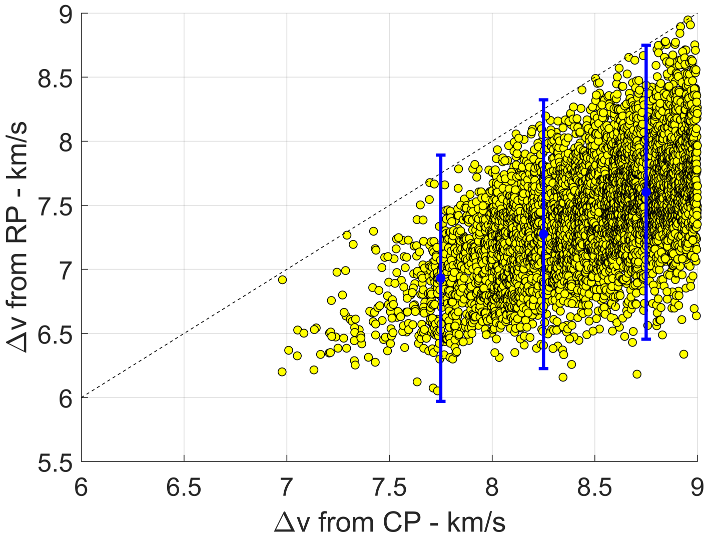


Figure 6.4: Summary of solutions as from RP and their estimates as from CP. Red error bars indicate 3σ distributions at 7.75, 8.25 and 8.75 km/s.

Figure 6.4 reports the Δv costs of an ensemble of unique asteroids tour solutions identified after exploring the baseline EM trajectory from Table 6.1 by means of both CP and RP sequential processes. Each point in Figure 6.4 represents a trajectory with 12 asteroids plus Mars with its Δv for the CP as indicated in the x axis, and the cost of the associated RP as indicated the y axis. The RP solutions are obtained by re-optimizing the solutions for Δv using MATLAB particle swarm optimization. The visiting epochs in vector y are allowed to vary on a ± 30 days-range with respect to the corresponding CP values. Figure 6.4 shows that the refinement of CP solutions may achieve some improvement of Δv (3σ distribution), but the transcribed problem still provides a rather accurate estimate of the goodness of the trajectory, without the need to deal with the continuous variable optimization.

Thus, the CP assumption already provides a very good estimate of the Δv of a given transfer. As additional proof, Figure 6.5, shows the Δv costs of the asteroid-to-asteroid

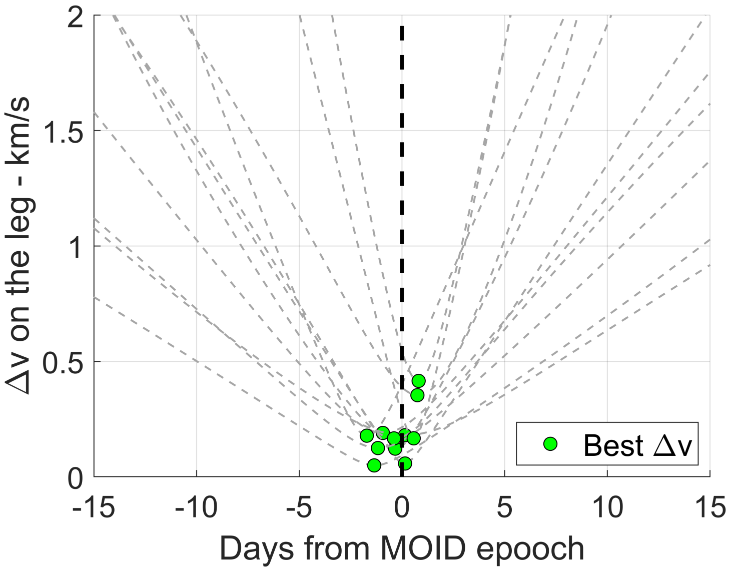
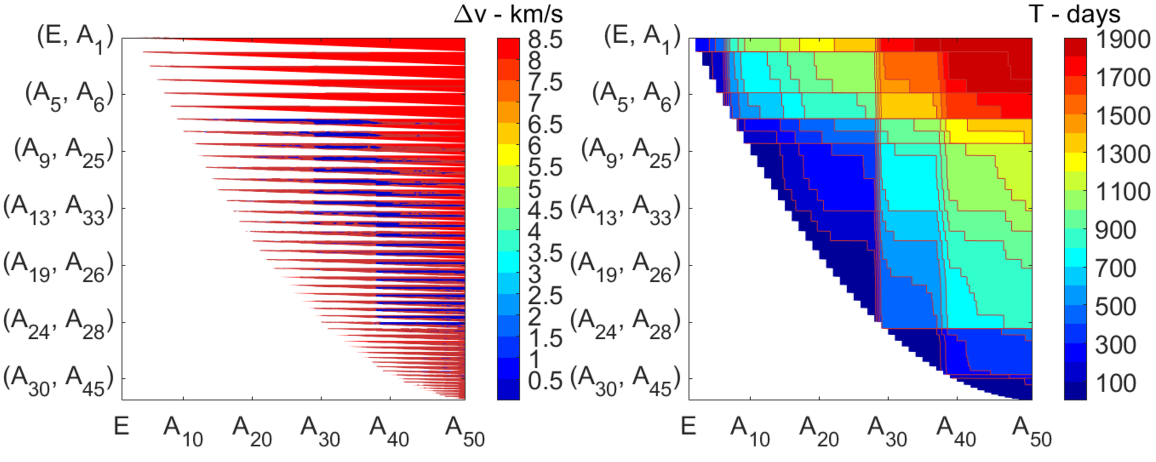


Figure 6.5: Δv variation on the different legs of the CASTAway EM reference trajectory with respect to the MOID epochs for each of the visited asteroids.

legs completing a full 12-asteroid tours in the MAB. The x-axis in Figure 6.5 describes the sensitivity of each asteroid-to-asteroid leg to changes on the asteroid fly-by epoch, as centred around each asteroid's MOID point crossing epoch. Circles highlight the actual minimum Δv possible for the given leg. The Figure illustrates that the described procedure results in a very efficient process to find a fixed time of flight between two consecutive objects. Note that the a-priori knowledge of the asteroid fly-by time was $y_1 < t_{fb} < y_1 + TOF$, and that the process allows to identify a fly-by epoch within few days of its optimal configuration.

6.4 Tree Searches on Transcribed Space

The structure of the transcribed space is similar to the one described in Chapter 5. This space is made by nodes that are couples of asteroids at their encounter epochs, that uniquely define the Lambert arc between them (for fixed number of revolutions). By referring again to Figure 5.2, to connect different nodes on this space, namely (P_i, P_j) and (P_j, P_k), one uses the $\Delta v_{ij} = |\vec{v}_j^+ - \vec{v}_j^-|$, where one has replaced $\vec{v}_{\infty,j}^-$ and $\vec{v}_{\infty,j}^+$ with solutions of the Lambert arcs $\vec{v}_j^-$ and $\vec{v}_j^+$, respectively. Therefore, for the triplet (P_i, P_j, P_k) one has a unique cost. Being the Δv_{ij} a tri-asteroids dependent cost, unique for each of the legs of the search space, a tri-structured score matrix can be created. Figure 6.6 shows an example of score matrix for the smallest set of 49 asteroids (see section 6.3.2). Each row, in y axis, represents a couple of asteroids, i.e., a trajectory between two objects, and each column, in x axis, is encoded with asteroids in the catalogue that completes the triplet. The third dimension is completed with Δv_{ij} for Figure 6.6.a and with time of flight T for Figure 6.6.b [2]. Asteroids and planets are ordered with respect to their visiting time. In fact, one should recall that each asteroid and planet comes with its own visiting epoch from the procedures described in sections 6.2 and 6.3. Thus, asteroid A_1 cannot be reached by asteroid A_5, since A_5 is encountered after A_1 (this also explains the white space in Figure 6.6).



(a)

(b)

Figure 6.6: Score matrix with Δv values (a) and time of flight T (b) for the catalogue of 49 asteroids ($d_{th} = 0.03$ AU).

Such score matrix allows to usefully exploit the substructure of unique triplets [3]. In fact, one can either pre-compute all the possible triplets in the score matrix or store the cost for each triplet at the first instance that is computed during the search process. In this way one identifies several characteristics of the whole set of asteroids (like the Δv as in Figure

[2] In this case, for example, Figure 6.6.b shows the time of flight impact that adding a given triplet has on the overall sequence.

[3] Figure 6.6 is for representation purposes. One can extract the transfers/asteroids that are feasibly reachable by simply pruning rows and columns from the matrix.

6.6.a, the transfer time as in Figure 6.6.b, encountering conditions, relative velocities, phase angles, and so on) without even addressing the combinatorial problem.

Having defined the search space as a graph of connected nodes, each of which defined as pairs of asteroids, it may then be tempting to consider complete tree traverse algorithms to solve the CSP problem, i.e. to find all the sequences of asteroids. DF and BF may allow to systematically cross the tree graph representing the entire search space of the problem (see section 2.1.4 describing such procedures). By means of the binomial coefficient (see Eq. 1.1), one can quickly see how the total number of possible paths in a tree quickly grows to unfeasible values. Table 6.2, for example, shows all the possible sequences of 12-asteroid-long paths that exist for each set identified for different distance threshold (see section 6.3.2). Considering pruning criteria derived from realistic mission design scenarios, one can still complete tree traverse explorations for the smallest catalogues in Table 6.2. Nevertheless, given the exponential growth of the number of combinations, the task quickly becomes an impossible endeavor for larger sets.

Table 6.2: Number of combinations of 12 asteroids sequences.

Number of asteroids	**Number of sequences with 12 asteroids**
49	9×10^{10}
98	8×10^{14}
158	3×10^{17}
562	2×10^{24}
1026	2×10^{27}

One might then want to consider incomplete searches like BS, on which the computational effort associated with tree exploration is bounded by capping the number of branches that can be expanded at any one level. More specifically, from all the branches generated at one level, only a limited set of them, referred as the beam, is selected to be expanded at successive levels. The dimension of the beam (i.e., the number of sequences to be kept in memory for further expansion) is referred as the beam width (BW). The sequence selection is usually done by prioritizing those that minimize a cost function, e.g., the Δv. Although this is a well-established procedure to obtain multiple locally optimal solutions for the problem at hand (see Chapter 2), the guaranty on global optimality with respect to a given criterion (e.g., the Δv) is not assured. Selecting the proper BW is thus a compromise between solution quality and number and computational effort.

Performances of the BS algorithm on the problem at hand are evaluated over a grid of settings for the BW for the different asteroids' catalogues considered. The BW varies from 0 to an arbitrary high number, e.g., 200,000, that allows to consistently find tens of thousands of solutions. Results are provided in Figure 6.7. The analysis in Figure 6.7 show several plateaus, with transitions at a priori unknown BW. One can first observe first that transitions to the same Δv level occur at a larger BW for larger catalogues: see for example the 98 and 158 asteroids catalogues, which both reach 6.977 km/s level, but the 158 asteroid catalogue requires more than twice the BW to reach it than for the 98 catalogue. This feature suggests an underlying multi-modal structure of rapidly increasing complexity. The last of these plateau transitions must thus coincide with the global optimum. However, one does not know a priori the BW necessary to reach this final transition.

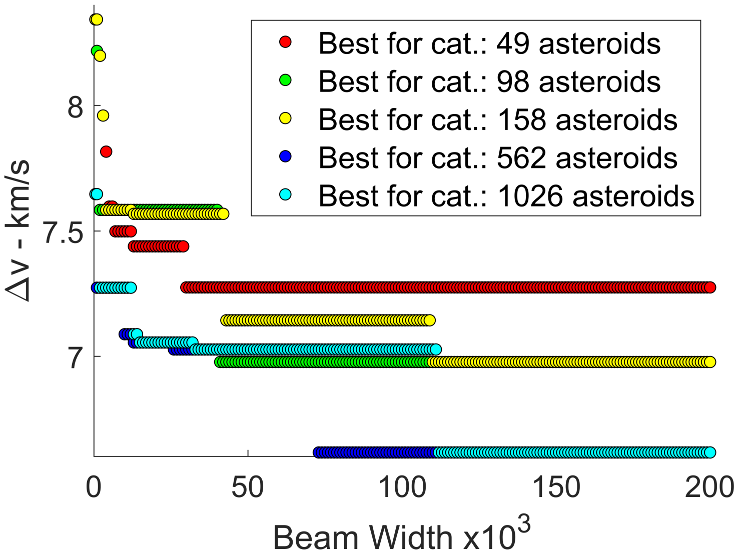


Figure 6.7: Best Δv solutions with respect to the BW for different catalogue considered.

6.5 Modified Dynamic Programming for Graph Exploration

As discussed above, as the number of asteroids in the dataset grows, the dimensionality of the problem quickly prevents the possibility to perform an exhaustive DF/BF exploration. Similarly to Chapter 5, efficient computational strategies, such as the single-objective variant of dynamic programming (SODP), may however mitigate this issue and enable a search guaranteeing the global optimality of the output of the search. For dynamic programming to be applicable, the problem needs to be decomposed into independent sub-problems, which are tackled separately, then the solutions of the sub-problems are combined to gradually solve the original problem. This has been done in section 6.3.2, showing that the solution of independent sub-problems for the asteroid tour mission tackled here corresponds to link a triplet of asteroids with Lambert arcs.

Beyond the aim of finding optimal asteroids tours from a single-objective point of view, the definition of feasible asteroids sequences for practical mission scenarios usually comes together with the definition of the scientific interest around specific (sequences of) asteroids. In theory, if one could find a 'science' function that is $f(X) = S$, where X encodes again a sequence of asteroids, and S the science that can be performed with the given sequence X, one could set-up, for example, a multi-objective optimization (like the MODP approach presented in Chapter 5) and optimize for different functions like the overall Δv (to be minimized) and S (to be maximized). However, the definition of such function $f(X) = S$ is not trivial and any function adopted (e.g., the ones employed in GTOC competitions [236] or in practical mission scenarios studies [9]) would be arbitrary and problem-dependent. In this sense, a CSP-like approach becomes relevant, as its aim is to find as many solutions as possible that satisfy given constraints, and (possibly) optimal solutions with respect to one or more criteria.

A simple modification to the SODP approach presented in Chapter 5 could be employed for this purpose. In SODP algorithm, if the graph expansion arrives at a node that is in common to two or more paths, then only the path with the minimum objective function is saved for further expansion and the other ones are discarded. This is because all the successive nodes that are attached to these solutions make the cost function vary of a constant value for all the paths that arrive at the common node.

The modification to the SODP scheme could be such that the best N solutions are retained alongside the single best solution at each node evaluation. It is thus possible to find and store rather than just the optimal solution, the best N solutions, where N is a user-defined parameter. Therefore, at termination of the sequence of asteroids instead of a single best solution, there are a set of N minimum solutions, as evaluated over data from start to end of the sequence. The interest in obtaining a set N of such solutions is that is diversified in their properties (the degree of diversification being dependent on how large N is). It should be noted that this modified approach is different from the MODP presented in Chapter 5 as it does not require multiple objective functions to be specified.

This modified approach lies somewhat in the middle between the SODP (one path saved per node evaluation) and MODP (multiple paths saved per node evaluation according to multiple criteria) approaches presented in Chapter 5. In this way, the global optimality is ensured for crucial and quantifiable functions (like the Δv) and diversity of solutions is preserved. This can be seen intuitively in Figure 6.8, on which an EVEMEJ trajectory is optimized with respect to the total Δv cost of the mission (i.e., the f_1 function as in Chapter 5) at different values of N. The figure shows the information available at different N in terms of the launch date (in x axis) and the total cost of the mission (y axis). As N grows, more solutions are included in the final set, that are diverse from one to the other (i.e., wider launch windows are captured). The global optimum is captured for all the scenarios.

The key question is how large N can be for such a process to be computationally efficient. This depends upon the number of solutions that can be retained in memory and kept for

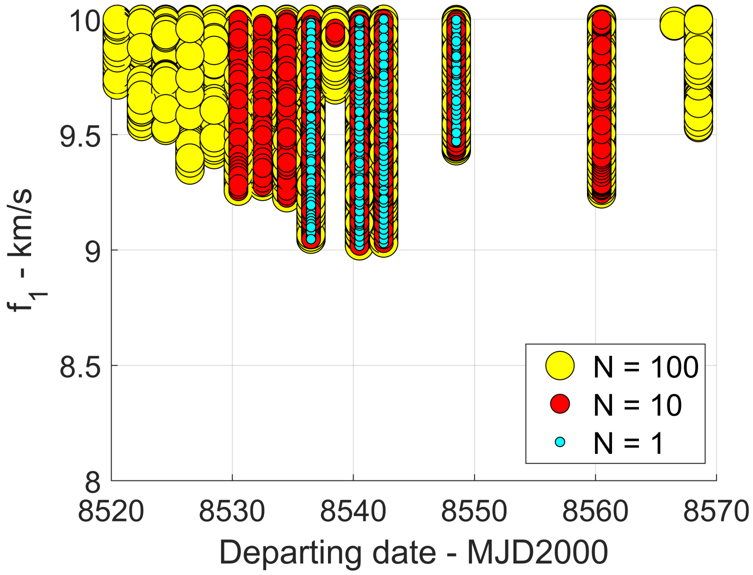


Figure 6.8: Departing dates and total cost for EVEMEJ sequence optimized with dynamic programming with different N.

further expansion. Similarly to well-known strategies like BS, the tuning of the parameter N is a compromise between solution number and computational effort.

As anticipated, the modified dynamic programming approach still carries out an optimization process, as the best N solutions are retained in terms of a specified objective function/ Δv, but retains more individuals within the population to be kept for further expansion, thus allowing more diversity to be included in the final set, that is similar to a CSP (see also Chapter 2).

6.6 Numerical results and discussion

Analyzing possible transfer scenarios relevant for MAB tour missions similar to CAST-Away requires a complete exploration of the search space, in terms of launch window analysis and transfer strategies. This implies a high number of MINLP/CSP problems to be solved, with associated challenges as described in previous sections. Therefore, a pipeline based on MGA assessment (section 6.2), transcription (section 6.3) and deterministic exploration (section 6.5) provides an efficient tool to explore the search space.

Applications of the pipeline are discussed in this section and are as follows:

- In section 6.6.1, the mission scenario is considered fixed (one single departing date and one single option in terms of gravity assist sequence are studied). The Earth-Mars baseline trajectory as presented in section 6.2 is explored with the modified dynamic programming approach and its efficiency is discussed. The number of asteroids is fixed to 12.

- In section 6.6.2, the last scenario considers a modified dynamic programming approach over the latest formulation of CASTAway mission design, for a submission to the latest ESA M-class call. In this scenario, according to the latest trajectory boundaries from ESA, the launch date can happen anytime within the 2037 year. The mission assumes a maximum number of asteroids fly-bys equal to 10 (the reduced number of asteroids compared to the first iteration of CASTAway design as in point (1) was mainly because of operational costs). The boundaries of the M-class call are then relaxed and a wider search within a 10-years launch window with free number of asteroids is also presented.

6.6.1 Earth-Mars Scenario

The scenario considered here assumes a fixed gravity assist baseline path and one single departing date from the Earth. The details of such trajectory are reported in Table 6.1. An analysis of modified dynamic programming approach is performed to assess the efficiency of the exploration. As in Chapter 5, to estimate the computational effort of such a scenario, one can consider the number of Lambert arcs to be solved (N_L), the number of manoeuvres to be computed (N_d) (either defects at the planets or at the asteroids), and the number of routes to be stored in memory (N_r). The five options for the d_{thr} are considered from section 6.3.2, namely 0.03 AU, 0.04 AU, 0.05 AU, 0.10 AU and 0.15 AU. Three options are considered for the parameter N, that are 1, 2, and 5 (again, one should recall that having N=1 corresponds to solve a SODP optimization). The analysis is completed with a comparison against the BS strategy to assess the pros and cons of the proposed exploration. The cost function to be analyzed is the overall Δv for the mission (i.e., f_1 in Eq. 5.2), with a cap at 5 km/s for the $v_{\infty,dep}$.

Table 6.3: Number of asteroids, Lambert problems and defects computed for different d_{thr} for the EM CASTAway scenario.

d_{thr}	**Number of asteroids**	N_L	N_d
0.03 AU	49	22100	20825
0.04 AU	98	166650	161700
0.05 AU	158	682640	669920
0.10 AU	562	29900930	29742164
0.15 AU	1026	181062154	180534276

Table 6.3 represents the computational effort in terms of N_L and N_d with respect to the d_{thr}. One should notice that these do not depend upon the selection of the dynamic programming parameter N, as dynamic programming does not avoid computing all the necessary Lambert arcs and defects to achieve global optimality (similarly to the cases considered in Chapter 5). In the largest catalogues considered, namely d_{thr}=0.10 AU and d_{thr}=0.15 AU with 562 and 1026 asteroids, the number of Lambert arcs and defects to be computed is quite high, leading to approximately 30 to 200 seconds to solve the whole problem (considering few microseconds for solving a single Lambert arc [250] on standard laptop (i.e., 4 GHz single core) on compiled code, as well as few nanoseconds for defects computation). However, one should notice that such MOID threshold are unnecessarily high for practical mission scenarios and will be in general lower than that. For example, MOID threshold of 0.05 AU (alongside an appropriate step size in launch dates considered, e.g. 5-10 days) allows a consistent scan of the space around the reference trajectory as shown in section 6.6.2.

Table 6.4: Number of routes N_r needed at each stage of the tree expansion with respect to the modified dynamic programming parameter N for different catalogues explored. The analysis is completed by running different beam searches for comparison.

Number of asteroids: 49			
Modified Dynamic Programming			
Tree Level	**N=1**	**N=2**	**N=5**
1	5	5	5
2	89	89	89
3	462	911	1731
4	623	1246	3101
5	600	1200	2999
6	489	978	2445
7	442	884	2210
8	353	706	1764
9	250	500	1245
10	198	396	972
11	113	212	502
12	198	331	612
Best Δv		7.274 km/s	
Beam Search			
BW	623	1246	3101
Number of solutions	0	0	0
Best Δv	$--$	$--$	$--$
Number of asteroids: 98			
Modified Dynamic Programming			
Tree Level	**N=1**	**N=2**	**N=5**
1	11	11	11
2	395	395	395
3	1643	3255	7956
4	2264	4528	11130
5	2204	4408	10706
6	1948	3896	9708
7	1735	3465	8572
8	1523	3041	7480
9	1204	2400	5890
10	818	1623	3967
11	652	1296	3182
12	1508	2871	6265
Best Δv		6.977 km/s	
Beam Search			
BW	2264	4528	11130
Number of solutions	25	61	140
Best Δv	7.583 km/s	7.583 km/s	7.5823 km/s
Number of asteroids: 158			
Modified Dynamic Programming			
Tree Level	**N=1**	**N=2**	**N=5**
1	14	14	14
2	755	755	755
3	4137	8211	20197

Continued on next page

Table 6.4 – *Continued from previous page*

4	5800	11600	28842
5	5583	11166	27740
6	5007	10014	25030
7	4437	8845	21915
8	3893	7774	19320
9	3119	6203	15274
10	2266	4504	11006
11	1644	3273	8117
12	6783	13266	31504
Best Δv		6.977 km/s	
Beam Search			
BW	5800	11600	28842
Number of solutions	82	301	626
Best Δv	7.583 km/s	7.583 km/s	7.5823 km/s
Number of asteroids: 562			
Modified Dynamic Programming			
Tree Level	**N=1**	**N=2**	**N=5**
1	59	59	59
2	6430	6430	6430
3	35095	68976	165422
4	58255	115091	280296
5	56175	112111	279088
6	55505	110920	275684
7	51091	100838	247895
8	43881	87130	214706
9	37136	73756	181295
10	27436	54065	131298
11	18931	37612	92863
12	121614	239063	562689
Best Δv		6.458 km/s	
Beam Search			
BW	58255	115091	280296
Number of solutions	7937	35772	80104
Best Δv	7.026 km/s	7.026 km/s	7.026 km/s
Number of asteroids: 1026			
Modified Dynamic Programming			
Tree Level	**N=1**	**N=2**	**N=5**
1	91	91	91
2	11758	11758	11758
3	79835	155661	372096
4	152678	302137	740700
5	152558	304259	756211
6	150604	300667	749770
7	141546	282185	699053
8	127398	251752	617577
9	105521	208110	507140
10	77213	151683	365084
11	51782	102322	249330
12	381195	727658	1654189
Best Δv		6.458 km/s	

Continued on next page

Table 6.4 – *Continued from previous page*

Beam Search			
BW	152678	304259	756211
Number of solutions	46639	69135	208984
Best Δv	6.616 km/s	6.616 km/s	6.616 km/s

Table 6.4 shows the number of routes N_r that are stored at each level of the tree expansion while exploring different catalogues with respect to the dynamic programming parameter N. A BS is also run with BW equal to the maximum number of trajectories needed by dynamic programming for each case considered. In this way, one assesses the advantages of employing the proposed approach over standard strategies by simply tuning a single parameter (i.e., N). From Table 6.4, one notices that, for the cases considered, the BS does not find the global optimum solution (in terms of Δv) and always finds more than one order of magnitude less solutions than the modified dynamic programming. It is interesting to notice that for the smallest catalogue considered (i.e., 49 asteroids) no solutions are found with the defined beam widths (the minimum beam width to obtain solutions for this case is 4,000). This is substantially higher than the maximum number of solutions required by dynamic programming, which is 623 for N=1. In other words, while not ensuring global optimality on the transcribed space, BS needs to evaluate more 12-asteroids sequences than dynamic programming to obtain the same order of magnitude of solutions.

In addition, the modified dynamic programming approach manages the information in a way that the solutions quality is better (in terms of best Δv) compared to the BS simulations considered here. This can be seen in Figure 6.9, that is obtained for the largest catalogue explored (1,026 asteroids) and the largest parameter setting employed (N=5 for dynamic programming and beam width of 757,000 for BS). From Figure 6.9, assuming to classify the number of solution from beam search and dynamic programming (at tree level 12) in different bands of Δv, one notices that dynamic programming finds again many more order of magnitude of solutions compared to the beam search. For example, dynamic programming finds two solutions in the 6-6.5 km/s band (one of which is the globally optimal one), while BS finds none. This ultimately implies that the solutions

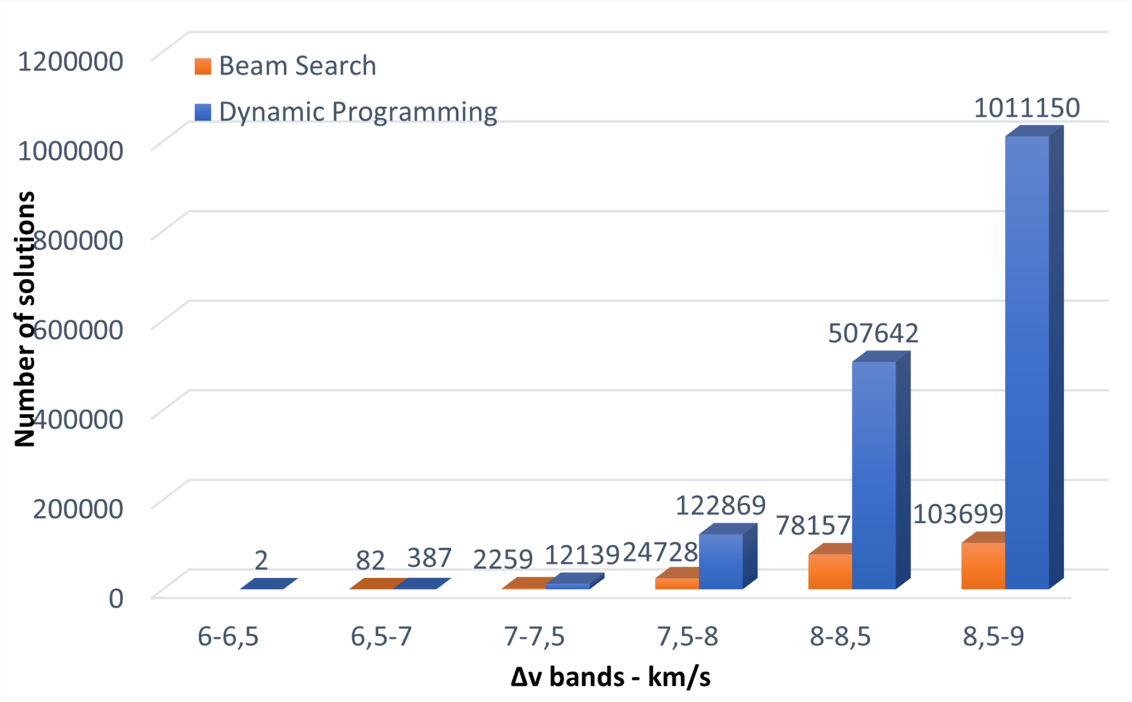


Figure 6.9: Number of solutions with respect to different Δv bands for beam search (beam width of 757,000) and dynamic programming (N=5) used on the 1026-asteroids catalogue.

quality is also improved overall (not only their number).

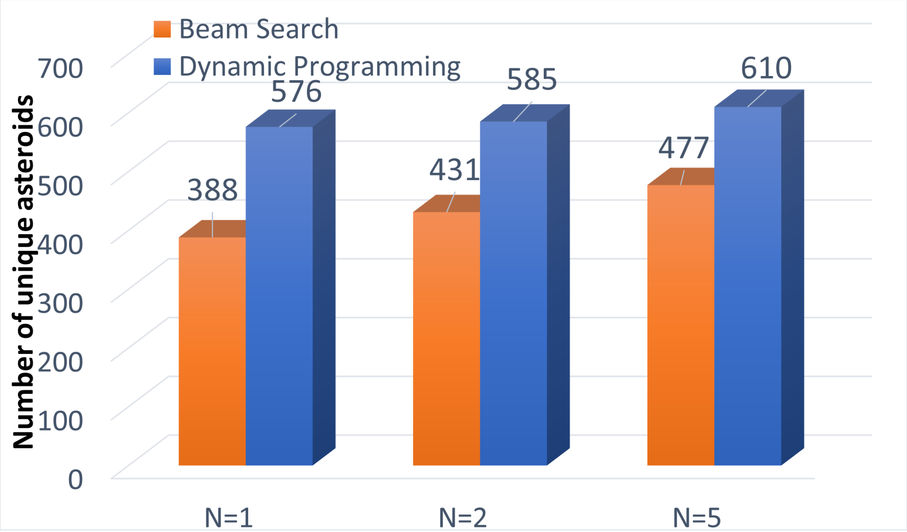


Figure 6.10: Number of unique asteroids within the final solution set for BS and dynamic programming at different N.

Finally, Figure 6.10 shows that the modified dynamic programming approach provides the highest asteroids' diversity in the final set of sequences, that is an analogous demonstration of Figure 6.8, on which the diversity grows with N, as expected. One notices that Figure 6.10 only represents the largest set explored, i.e., 1026 asteroids, but this behavior is experienced for all the catalogues explored.

To sum up, the following points have been demonstrated:

- Global optimality on the transcribed space is assured with dynamic programming approaches, while commonly-used heuristics do not guarantee it.
- No parameters need to be tuned (e.g., there is no BW to be selected). A simple set-up of N=1 already provides the global optimality with the minimum computational effort. Setting N>1 simply allows to seek for higher number of sequences in the final solution set [4].
- To obtain the same number of solutions, BS-like approaches generally needs more sequences calculations, typically 1 to 2 orders of magnitude more.
- On the cases considered, dynamic programming allows for better solution quality in terms of Δv and higher diversity.

As anticipated, the main drawback of the modified dynamic programming is the unpredictability of the maximum number of solutions that will be needed during the search. As expected, such number grows with the parameter N (and obviously with the number of asteroids in the catalogue). However, two considerations can be done:

- The maximum number of solutions for most practical cases, e.g., d_{thr} between 0.05 AU and 0.1 AU (see also section 6.6.2), is usually manageable as from Table 6.4.
- One can always hybridize the modified dynamic programming with a BS-like pruning (thus introducing the beam width) to keep a user-defined maximum number of solutions, alongside the globally optimal path.

[4] The final set-up on N>1 cases will always depend upon the working environment. If the machine/environment allows for high RAM, then N>>1 can be used.

6.6.2 CASTAway Mission Design

CASTAway trajectory design, defined as the exploration of mission opportunities for ESA's M7 mission call, is here used as a further stress test case to show the versatility of the presented tools in designing complex asteroids' tours. Table 6.5 summarizes the boundary conditions for the mission call. The pre-pruned main belt asteroid database contains approximately 102,000 objects, as described in section 6.3.1.

Table 6.5: Boundary conditions for CASTAway mission compatible with M7 call.

Value	Description	Constraints
t_0	Launch epoch	Jan. 1, 2037 - Dec. 31, 2037
$v_{\infty,dep}$	Earth escape velocity	$v_\infty \leq 5$ km/s
TOF	Overall mission duration	$TOF \leq 7$ years
N_{asts}	Number of visited asteroids	$N_{asts} = 10$

To generate baseline MGA trajectories to access the MAB region, planetary sequences are analysed that include all permutations of Earth, Venus, and Mars, with up to 2 planetary swing-bys. The 2037 launch window is discretized in 75 departing dates (approximately 5 days' time step). It should be noted that, given the launch date time-step considered, a $d_{thr} = 0.05$ AU ensures a sustained sweep of the MAB, exploring consistently all the reachable space. In theory, larger step size could also be allowed (e.g., up to 10 days), as shown in Figure 6.11, at the price of a smaller number of options to be optimized in the combinatorial step, but with larger $d_{thr} = 0.1$ AU.

Figure 6.12 shows the ratio between the time spent in the MAB region Δt_{MAB} and the overall Δv consumption needed for each baseline trajectory analysed. This allows to identify the most promising MGA options to explore the MAB region, as these are characterized by the highest $\Delta t_{MAB}/\Delta v$ ratio. As a general understanding of the results, the launch window analysis indicates that opportunities to explore the MAB are available through the entire launch window. A direct launch from Earth is assumed to occur with infinity velocity equal to 5 km/s for all the launch date scanned. As expected, a direct escape from the Earth towards the MAB is less sensitive to launch date, at the cost of shallower access to the MAB with respect to MGA strategies. Among all the MGA strategies considered

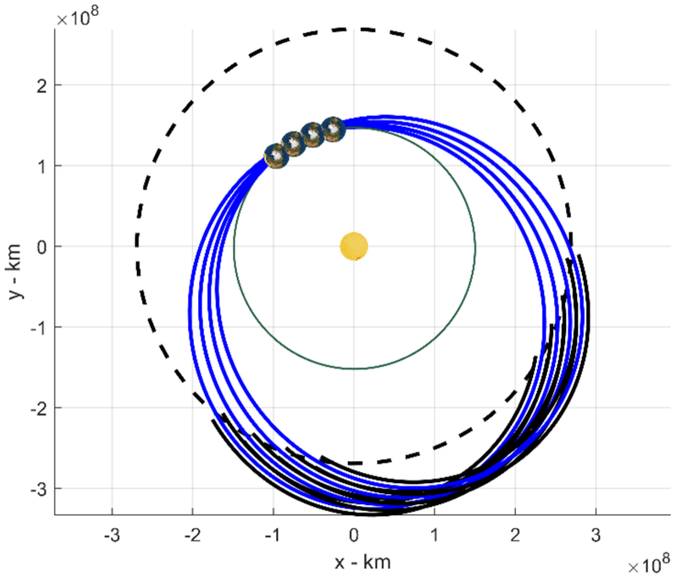
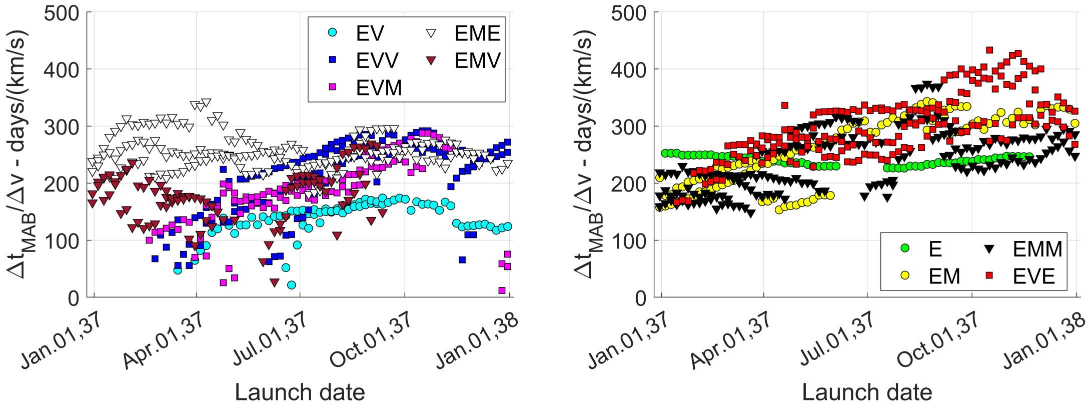


Figure 6.11: Reference trajectories for a direct launch from Earth (blue lines). Black lines, centered in reference trajectories and distant 0.05 AU from them, represent the space within which asteroid catalogues are built to explore asteroid tour opportunities. Earth departures are spaced by 10 days. Dotted lines are the limits of the MAB. Sun and planets textures are not in scale.

here, employing a single fly-by (i.e., for EV and EM) allows to reduce the overall Δv to reach the MAB, as no phasing manoeuvres are needed to encounter a third planet.

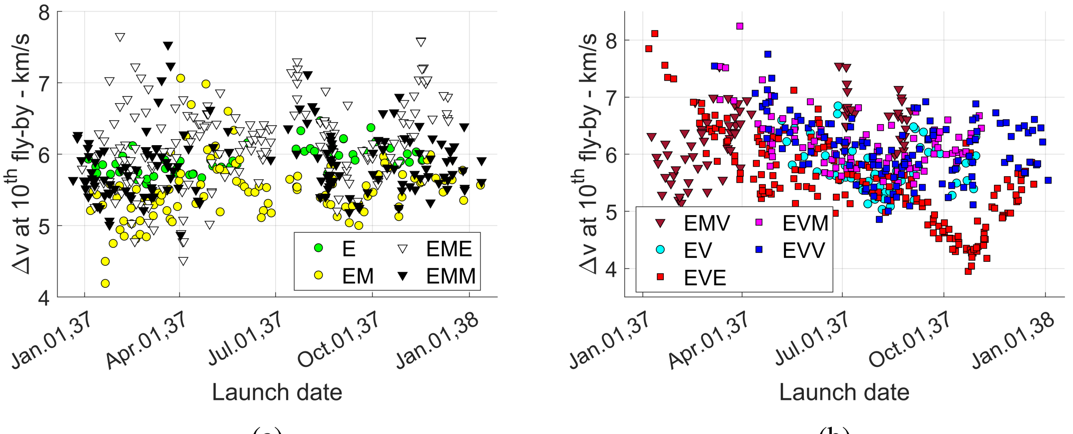


(a) (b)

Figure 6.12: Time accumulated in the MAB per total Δv with respect to the launch date for different strategies: E, EM, EMM, EVE (a), and EV, EVV, EVM, EME, EMV (b).

Between EV an EM, as expected, the latter performs better (i.e., higher $\Delta t_{MAB}/\Delta v$ ratio) in terms of MAB exploration as the spacecraft is launched directly towards high-apoapsis regions. When considering strategies with two fly-bys, EVE provides the highest $\Delta t_{MAB}/\Delta v$

ratio in the region around November and December 2037. This is because the more favourable phasing conditions in terms of Δv between Earth and Venus when compared with strategies with Mars. Strategies like EMM, EME and EVV are quite competitive in terms of Δt_{MAB} spent in the MAB, however, they come with generally higher Δv consumption due to the required phasing conditions between Earth, Venus, and Mars, thus making them less favourable when compared to single-flyby strategies as EM.



(a) (b)

Figure 6.13: Launch window analysis for 2037 launch opportunities for CASTAway-M7. Different gravity assist options are shown E, EM, EME, EMM (a) and EMV, EV, EVE, EVM, EVV (b).

Results (after refinement) are shown in Figure 6.13, showing the best opportunities for each gravity assist option during the 2037. One notices that Figure 6.13 is qualitatively in good accordance with Figure 6.12 (e.g., see the EVE strategy), as an additional proof of the effectiveness of the proposed pipeline. Figure 6.14 represents the trajectory followed by the spacecraft for the Δv-optimal EVE sequence (overall Δv = 3.951 km/s).

The modified dynamic programming (N=5) is able to provide a wide catalogue of trajectories for CASTAway, in the order of millions of solutions for each MGA option considered, as shown in Figure 6.15 [5]. Such high number of trajectories constitute a database that can be accessed by in successive stages of the design for further analysis by scientists, guidance analysts and so on. For example, one can post-process the solutions with respect to

[5] Again, one notices that the choice N=5 is just an example, and one is free to select the proper N>1 set-up (if needed) according to the power available on the personal machine/environment.

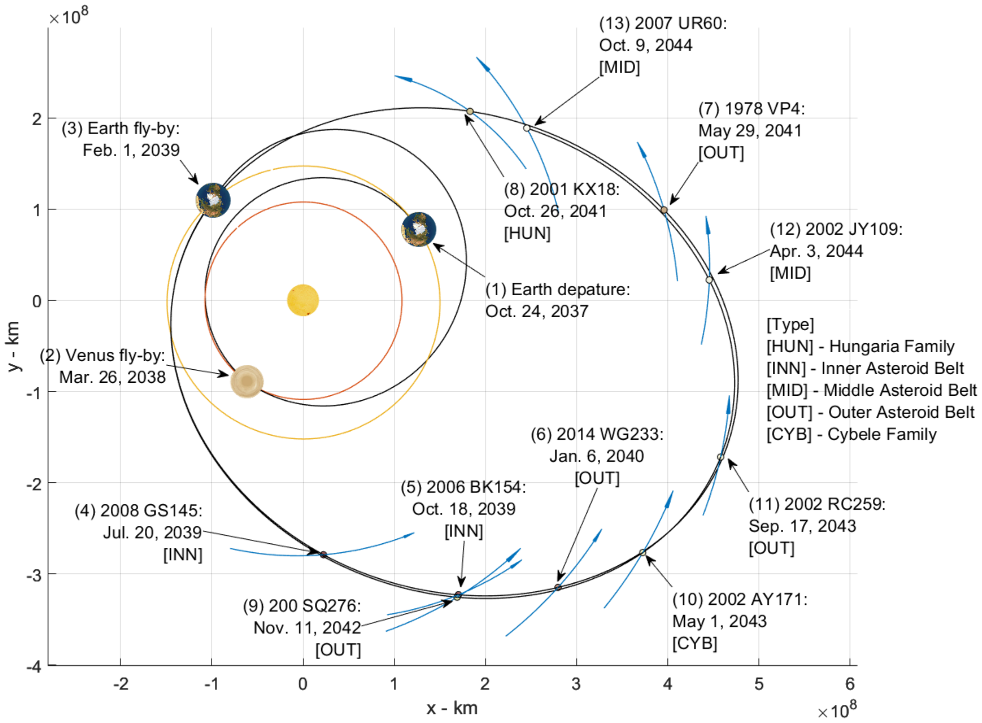


Figure 6.14: EVE strategy for CASTAway-M7 call.

scientifically interesting criteria like [9]:

- Criterion 1: scientifically compelling asteroids to be visited [6]. See also Appendix E.
- Criterion 2: number of asteroids that have diameter higher than a threshold (e.g., 10 km)
- Criterion 3: number of asteroids that have information on spectral type
- Criterion 4: number of regions visited (i.e., Hungaria, Inner Belt, Middle Belt, Outer Belt, Cybele, Hylda) [9]
- Criterion 5: number of asteroids families visited (i.e., Flora, Vesta, Nysa, Maria, Eunomia, Geflon, Koronis, Eos, Thermis, Hygiea) [9]. See also Appendix E.

For example, in the year 2037, if one orders the number of sequences with respect to such criteria (i.e., first the sequences that maximize successively Criterion 1, 2, and so on), one

[6] The author thanks Professor C. Snodgrass for providing the list of scientifically interesting asteroids.

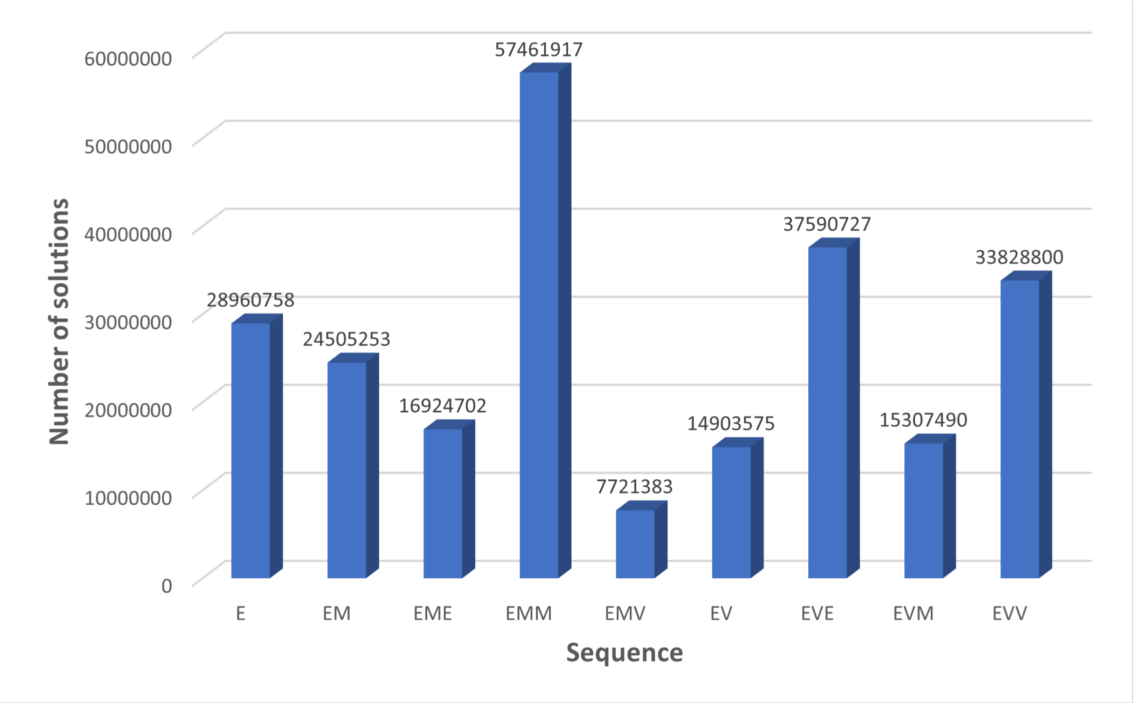


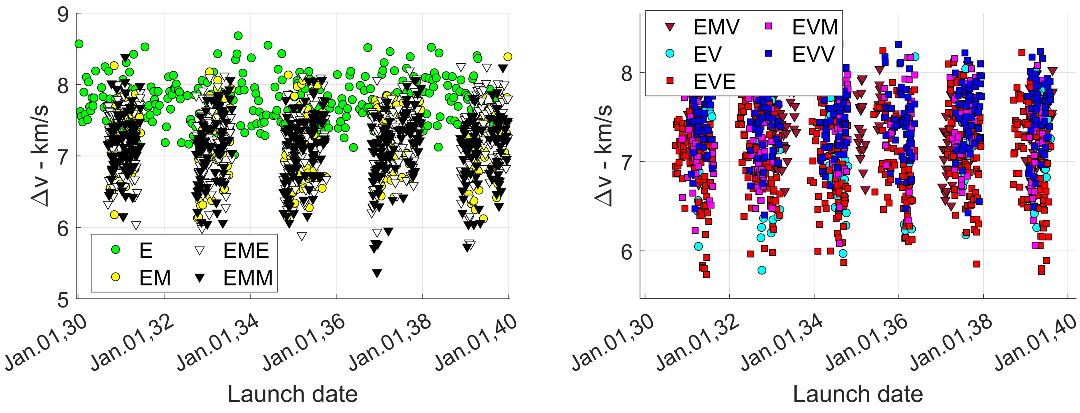
Figure 6.15: Number of solutions that satisfy the constraints for the CASTAway-M7 mission design.

can classify the MGA options as in Table 6.6. The Table shows the characteristics of the sequence that maximizes the number of asteroids for each Criterion. It can be noticed that the maximum number of special asteroids that are visited in a sequence of 10 objects is 2. EM scenario has a good balance between all the criteria, while EVE allows for the highest number of regions visited, due to the high apoapsis reachable after the Earth fly-by. Surprisingly, EMV allows for the highest number of asteroids families visited (12) while EVV maximizes the number of asteroids with diameter greater than 10 km/s.

Table 6.6: Number of asteroids satisfying each criterion for different MGA options.

Option	Criterion 1	Criterion 2	Criterion 3	Criterion 4	Criterion 5
E	1	4	1	3	8
EM	2	2	2	3	7
EME	1	3	0	5	6
EMM	2	1	1	4	8
EMV	1	4	0	4	12
EV	2	1	0	3	5
EVE	2	1	1	5	9
EVM	1	3	1	4	8
EVV	1	6	4	4	6

The most diverse exploration in terms of asteroids regions is provided by EME and EVE sequence, that allows to cross up to 5 different orbital regions. The representation of such trajectory and the regions visited is shown in Figure 1.1 and Figure 6.14. In fact, one notices here that the EVE strategy selected for CASTAway submission maximises the number of asteroids' regions visited.



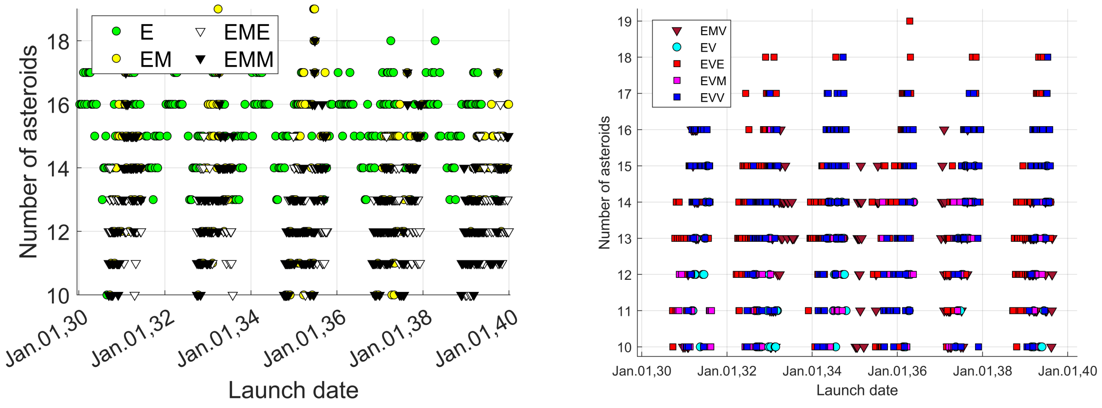
(a)

(b)

Figure 6.16: Launch window analysis for 2030-2040 launch opportunities for CASTAway extended mission design. Different gravity assist options are shown: E, EM, EME, EMM (a) and EMV, EV, EVE, EVM, EVV (b).

It should be noted again that CASTAway M7 mission design considers very strict boundary conditions from M7 call, e.g. on the number of asteroids to be visited, or the launch window. In theory, as already demonstrated in Chapter 5, the pipeline presented allows for much wider search spaces to be explored, both in terms of number of asteroids to be visited and the dimension of the launch year. Figure 6.16 provides results for an exploration like the scenario in Table 6.5, but with the following modifications: (1) the number of asteroids is maximized during the search; (2) the launch can happen anytime between 2030 to 2040.

Figure 6.16 shows that visiting the MAB with almost any sequence of gravity assist body is not a rare event. However, as expected, the Earth provides more flexibility in the launch date, but at a higher cost overall, while the sequences employing gravity assists have better solutions in terms of optimal Δv, but are spaced one to the other of a synodic



(a) (b)

Figure 6.17: Number of asteroids for the launch window analysis in the 2030-2040 time-frame. Different gravity assist options are shown: E, EM, EME, EMM (a) and EMV, EV, EVE, EVM, EVV (b).

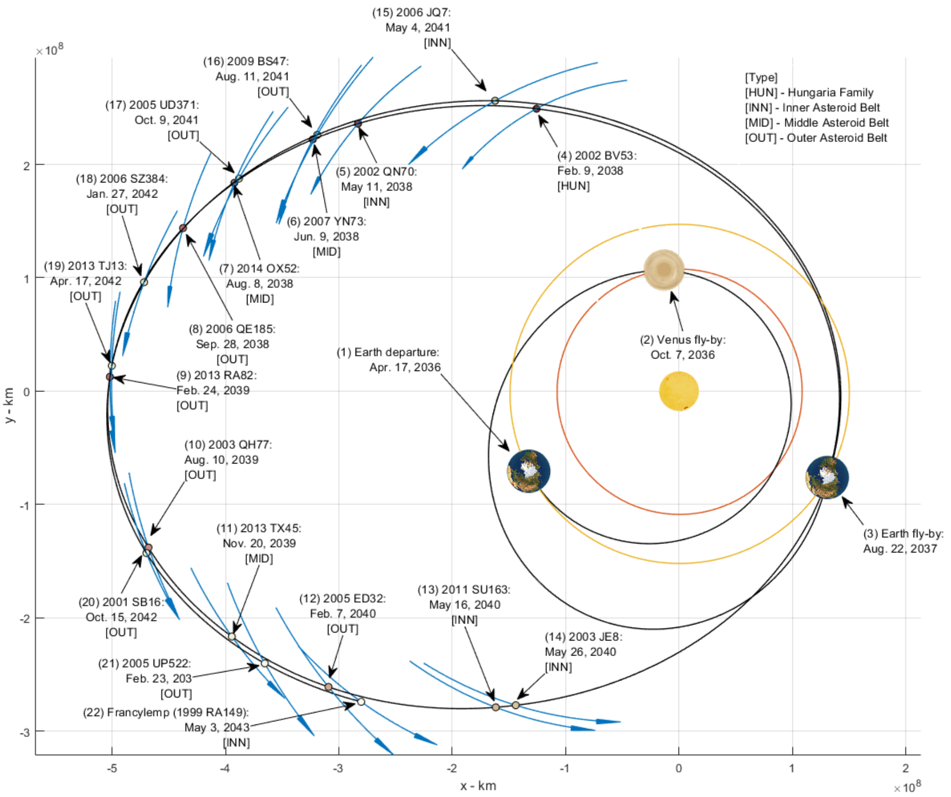


Figure 6.18: EVE strategy to visit 19 asteroids.

period. Figure 6.17 shows the maximum number of asteroids that are reachable with different gravity assist sequences in the 2030-2040 time-frame, which are compatible with the scenario under consideration. As expected, EM and EVE provide the highest number (19). Figure 6.18 shows the EVE trajectory for the longest asteroids-sequence found (overall Δv = 7.787 km/s).

6.7 Conclusions

This Chapter has tackled the asteroid tour problem with gravity assist manoeuvres from two different, though related, formulations: global optimization problem, aiming at finding the global optimum solution with respect to a cost function (i.e., overall Δv of the mission), and a formulation based on a paradigm inspired by the CSP, aiming at finding as many solutions as possible satisfying given mission-driven constraints.

A pipeline to perform preliminary mission design of missions that visit multiple asteroids has been presented. The crucial step of such pipeline is the transcription process employed to transform the mixed-integer problem into a combinatorial one. This opened the possibility of applying dynamic programming principles and to extend them to retain multiple diverse solutions. In this way, the pipeline is versatile enough to:

- Guarantee global optimality on the transcribed space.
- Obtain more and diverse solutions with respect to standard methods like BS, by managing the search conveniently.
- Obtain better-quality solutions in terms of cost function (Δv), since they are searched around the optimal ones.
- Explore very wide search spaces in terms of number of asteroids visited (up to 19) and launch window analysis (10 years explored).

The proposed process is ideal to support preliminary mission design studies of asteroid exploration missions such as CASTAway or MANTIS. Indeed, the process was implemented to support CASTAway mission proposal submission to the European Space Agency's 7th Medium-size call, enabling a comprehensive main-belt exploration opportunities for a launch window in 2037.

Chapter 7

Conclusions

The thesis presented a pipeline to tackle mixed-integer space trajectory design problems robustly and efficiently. The versatility of the approaches has also been shown on different test cases of complex configuration. This last Chapter draws the conclusions of the present study. After presenting the summary of the work in section 7.1, section 7.2 discusses the response to the objectives and aim declared in Chapter 1. Section 7.3 finally discusses current limitations and possible paths for further research.

7.1 Summary of the Work

The mission analysis of multi-target space missions, belonging to the so-called class of mixed-integer design problems, is studied in the present work.

After having contextualized the thesis within the current literature (Chapter 2) and having presented the mathematical framework (Chapter 3), the thesis focused on the development of a multi-fidelity pipeline to efficiently and robustly planning interplanetary space missions that aim to visit multiple targets. The efficiency is achieved by exploiting search space structures at each fidelity level, while robustness is assured with the in-depth discussion on information that can be passed from one fidelity level to another.

Particularly, the first level starts with a Tisserand map exploration (Chapter 4). This allows to assess the feasibility of different fly-by sequences, even for very complex transfers, requiring more than 40 fly-bys (see Appendix D). Compared to current literature, Tisserand map exploration has been extended to missions that aim to maximise the final orbital inclination by means of resonant transfers. Additionally, it has been shown how the information from Tisserand maps can be exploited to speed up successive stages of mission planning.

At the next level of fidelity (Chapter 5), one looks for actual dates and transfer times between the selected objects. A model based upon Δv defects allows to transcribe the search space into a graph of interconnected nodes. In this way, dynamic programming can be easily applied, that allows to obtain globally optimal paths with limited computational effort. Compared to current literature, dynamic programming principles have been extended to deal with the challenging multi-objective optimization of interplanetary trajectories, thus allowing to obtain optimal Pareto fronts on complex mission scenarios.

Finally, within the context of missions to visit multiple asteroids (Chapter 6), dynamic programming principles have been further extended. The resulting approach, called modified dynamic programming, allows to retain multiple paths at each object-to-object evaluation. Compared to current standard methods employed in literature, it allows to obtain more, diverse and better-quality solutions, without preventing the globally optimal path. In this sense, the optimization is enhanced by retaining multiple sub-optimal solutions that allow truthful representation of the topology of the search space. Again, stress test cases of complex configuration have been used to prove the adaptability of the proposed methodology.

7.2 Fulfilled Objectives and Response to Aim

In Chapter 1 five objectives are declared. The assessment of such objectives is here discussed, allowing to respond to the overall aim of the thesis.

Objective 1: ***To review the current trends for space mission analysis in the context of multi-target trajectory design, and relate such context to optimization and constraint satisfaction fields.***

Such objective has been thoroughly assessed in Chapter 2, where trajectory design approaches are discussed, and novelties with respect to literature are presented. In particular, it has been identified that most of the current literature tackles the design of mixed-integer interplanetary missions by formulating a global optimization problem, especially from the single-objective point of view. However, the more challenging multi-objective optimization is to be tackled as well, to properly inform the mission design with full extent of trajectory options at preliminary design stages. Moreover, a constraint satisfaction problem is also relevant, especially when multiple trajectory options are to be sought, that share similar performances in terms of Δv or mission duration, but are different with respect to scientific-related criteria, that are difficult to model.

Objective 2: ***To assess the feasibility of different model types on an overall multi-fidelity design of multi-target trajectories.***

The contributions to the second objective are found in Chapters 3, 4, 5 and 6. In particular, when presenting the mathematical framework, Chapter 3 presented an extension to the well known MGA-DSM model to include multiple DSMs on a same planet-to-planet leg. The resulting model, referred to as MGA-nDSM, better helps trajectory designers to plan missions where multiple DSMs are needed. Although not explicitly employed in Chapter 5 and 6, Chapter 3 has shown how it can be used to plan complex missions like those to Mercury in conjunction with the material provided in successive Chapters.

In Chapter 4, section 4.4 is dedicated to the assessment of the information that can be exchanged between levels of different fidelities. It has been noticed that the knowledge on infinity velocities at planetary encounters are useful to constraint the search for trajectories at the next fidelity level. This helps reducing the computational effort in terms of Lambert arc solved and defects computed, while preserving the Pareto front characteristics. On the other hand, any other information, e.g. about transfer times, should be taken with care, as the underlying assumptions on Tisserand map (i.e., circular co-planar planetary orbits) prevent accurate estimations on such critical mission parameter.

The robustness of the transcription process at the next fidelity level is ensured in Chapter 5 in section 5.4 by evaluating the effect of inserting a DSM to remove a Δv defect at a planetary encounter. The Chapter has demonstrated the relationships between such manoeuvres and the defects via a simple STM approach. In particular, the transcription process implied by the application of defects preserves the search space characteristics, thus allowing to capture optimal solutions with tree-traverse techniques like dynamic programming.

Finally, section 6.3.3 of Chapter 6 discusses that also the transcription process implied by the MOID search is able to provide accurate results even before the final refinement. Specifically, the refined solutions lie close within few days to their corresponding estimates at the combinatorial level.

Objective 3: ***To propose efficient strategies for the global exploration of the search space for multi-target missions.***

This objective is assessed in Chapters 4, 5 and 6. The exploration of meaningful search space is driven by the definition of nodes that allow tree-traverse techniques to be conveniently employed.

In the context of Tisserand map exploration in Chapter 4, a node is made by a triplet of (v_∞, α, k), that uniquely identifies an orbit on the Tisserand map. Such nodes are conveniently linked by fly-bys, VILTs and resonant transfers on an overall MGA sequence. The computational efficiency of Tisserand map exploration lies on the pre-computation of all the possible intersections and/or VILTs as in Algorithms 1 and 2, reducing the overall search of MGA paths to matrix manipulations. This strategy can also be adopted in successive stages, e.g., when solving Lambert arcs between different objects (see Chapters 5 and 6). In this case, a node is made by two objects an their specific encountering epochs.

This allows to pre-compute all the possible Lambert arcs and defects, thus permitting to construct a score matrix that encodes all the information on the search space (e.g., Δv, v_∞, phase angles, transfer times, and so on), without even solving the combinatorial problem.

The clever definition of such nodes allows dynamic programming to be applicable. In this way, optimal solutions are guaranteed on the graph space with limited computational effort.

Objective 4: ***To tackle the constraint satisfaction problem for multi-target missions.***

This objective has been assessed mainly in Chapter 6. One has defined the constraint satisfaction problem as the problem to find as many solutions as possible (i.e., compatibly with memory issues of the working environment) within given mission constraints, and possibly an optimal one with respect to a specific objective.

A modified dynamic programming approach has been proposed to answer such need. The governing parameter of the approach is N, defining the number of paths to be stored in memory at each node evaluation. This is the only tuning parameter that is needed and, compared to literature approaches, its tuning does not prevent optimal solutions to be found. In other words, $N = 1$ already corresponds to a SODP approach, thus guaranteeing global optimality at least for one objective, and the choice of $N > 1$ only allows to find more solutions.

Objective 5: ***To demonstrate the efficiency and robustness of such approaches with relevant and practical test cases.***

This last objective is assessed in Chapters 5 and 6, specifically in the numerical results sections 5.5 and 6.6. It has been shown that multi-objective optimization and modified dynamic programming approaches allow to plan MGA trajectories on complex mission scenarios with moderate effort.

In fact, apart from well-known mission options towards Jupiter and Saturn, the thesis addressed the design of novel and complex scenarios. Going beyond the current literature, the increased complexity of such scenarios lies on the trajectory planning for long MGA sequences and very large search spaces (10 years launch window). Such studies are ideal for planning future missions and to support preliminary trajectory design studies as those performed in the context of ESA F/M-class calls (see section 1.5.5).

To conclude, in Chapter 1, the following aim for the whole work is declared:

> *To establish an efficient and robust pipeline to tackle both the global optimization problem and the constraint satisfaction problem for space missions that aim to visit multiple targets.*

The aim response is considered to be achieved as all the objectives have been answered. The thesis thus contributed to current knowledge on multi-target space mission design by proposing multi-fidelity pipelines based on dynamic programming for finding optimal trajectories with limited computational effort.

7.3 Current Limitations and Future Work

Further implementations and possible research directions can be identified based on current limitations of the proposed work. Three main paths are identified.

The first one relates with handling multi-revolutions transfers on Lambert grids evaluation, when same planet-to-planet transfers are considered. There exist non-resonant ballistic trajectories that arrive at the same body with the same infinity velocity. Such transfers imply non-integer ratio of mean motions such that the return occurs at the alternative planet orbit crossing event. These transfers must involve co-planar orbits. These orbit solutions may be derived analytically. For a given initial crossing, a range of returning orbit solutions are possible corresponding to a bounded range of post fly-by semi-major axes (and consequently eccentricity). This case can be obtained with evaluation of a grid of Lambert arcs, and it is implemented in this way in the whole pipeline presented in the thesis. However, future versions may be adapted to include the option to substitute the analytical non-integer resonance to remove the need for evaluation of a Lambert grid for that leg, thus potentially reducing computational effort. This becomes particularly useful when dealing with small bodies (e.g., Saturn moons) that don't have as much flyby deflection power as planets. As additional improvement, one can add further type of resonance where an integer ratio of 'half' revolutions applies. In this case, the return occurs at the alternative orbit node of the spacecraft's orbit with respect to the planet orbit plane. In this case co-planarity is not required but the node constraint imposes a constraint on eccentricity. They may also be derived analytically.

The second improvement relates with managing the exploration with modified dynamic programming approach when addressing asteroid-related constraint satisfaction problem. As said, the tree exploration is governed by the parameter *N*, and larger values of *N* allow to capture more sequences and thus to preserve more information about the topography of the search space when compared to literature approaches. However, information about how much of the search space has been explored (i.e., how many diverse sequences have been obtained) can only be done as a post-processing step. In other words, during the tree search there is no explicit bias on how much adding a specific node would contribute to the diversity of the final solutions set, but rather a post-processing assessment of the solutions with respect to mission-specific criteria. This is a common issue of current tree-search and meta-heuristic strategies from literature, thus further research is needed. A possible solution that has been opened in [221] under the author's supervision is: (1) to explicitly define a diversity measure to be used in the search step and, (2) to use modified dynamic programming approach biasing the exploration not only by optimality but also by diversity.

Finally, a third research direction concerns the fidelity of the models employed when traversing the transcribed space. A truthful transcription is crucial to preserve the search space characteristics between models of different fidelities. As anticipated in Chapters 1 and 2, the application of deep neural networks are proving their efficiency to truthfully approximate critical mission parameters like Δv and transfer times with low computational effort. Such networks are trained over databases of transfers with the aim of reconstructing the processes to obtain them. These networks can then be used as approximators for optimal trajectories when exploring the tree graph. To the best knowledge of the author, there is still little literature on how to use such approach on multiple fly-by missions with DSMs or low-thrust transfers. The main issue with fly-by missions is, once more, the tri-dependency Δv cost of connecting two nodes. Future research can thus focus on developing deep neural networks as approximators for MGA missions with DSMs to be included into the presented dynamic programming approaches. The advantage could be that more complex dynamical models that go beyond Keplerian dynamics are tackled at early stages of the design. The ultimate aim is to have closer correspondence between models of different fidelities.

Appendix A

Lambert's Problem

This Appendix provides an overview about the of Lambert's problem and its solution. Since Lambert's problem is one of the most extensively studied problems in astrodynamics, the aim of this Appendix is just to provide a general understanding about it and bits of possible solution implementations. Section A.1 describes the problem and the variables that are needed to solve it, while section A.2 provides some information about how to solve it.

A.1 Description of the problem

In Lambert's problem, also known as the orbital two-point boundary value problem, the following quantities are given: (1) spacecraft initial and final positions, $\vec{r}_i$ and $\vec{r}_f$, respectively, and (2) the transfer time $\Delta t = t_f - t_i$ between the two. Solving the problem consists in finding the arc that connects $\vec{r}_i$ and $\vec{r}_f$ in the given Δt. This ultimately corresponds to find the velocities $\vec{v}_i$ and $\vec{v}_f$ at the positions $\vec{r}_i$ and $\vec{r}_f$, respectively, such that the state $[\vec{r}_i, \vec{v}_i]$ propagated by Δt leads to the final state $\left[\vec{r}_f, \vec{v}_f\right]$. A representation of the problem is given in Figure A.1.

From the Figure, the angle $\Delta\theta$ between $\vec{r}_i$ and $\vec{r}_f$ defines the direction of motion and it is

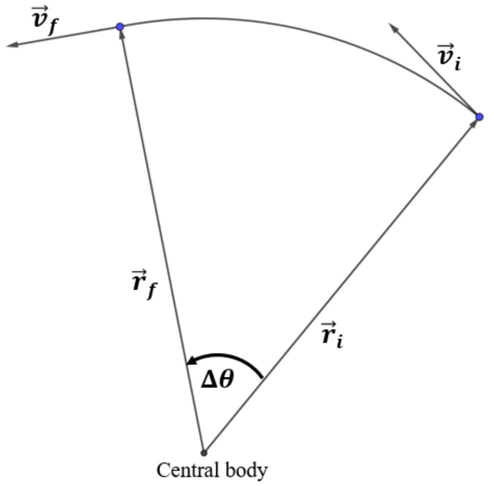


Figure A.1: Geometry of different ellipses for multi-revolution transfers.

defined as:

$$\Delta\theta = \arccos\left(\frac{\vec{r}_i \cdot \vec{r}_f}{r_i r_f}\right) \tag{A.1}$$

When implementing a solution to Lambert's problem, a boolean variable is required to select the appropriate value for $\Delta\theta$ that distinguished between two cases: (1) $\Delta\theta > 0$, i.e., the solution orbit has an inclination $i < 180$ degrees (prograde orbit), (2) $\Delta\theta < 0$, i.e., the solution orbit has an inclination $i > 180$ degrees (retrograde orbit).

If the transfer time Δt is sufficiently high, multiple solutions to the problem appear associated to different number of spacecraft revolutions about the main body. The maximum number of revolutions is given by:

$$N_{max} = floor\left(\frac{\Delta t}{2\pi\sqrt{\mu/a_m^3}}\right) \tag{A.2}$$

where $a_m = 1/4\left(r_i + r_i + |\vec{r}_i - \vec{r}_f|\right)$ is the semi-major axis of the minimum energy ellipse that connects $\vec{r}_i$ and $\vec{r}_f$ [38].

In the case of multiple revolutions, the geometry of the possible options are given in

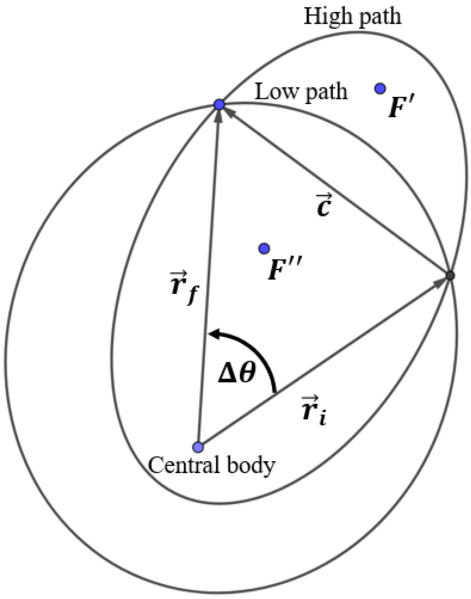


Figure A.2: Geometry of different ellipses for multi-revolution transfers.

Figure A.2. From the Figure, the vector $\vec{r}_f - \vec{r}_i$ is the chord line. In this case, another boolean variable should be set to select between two solutions: (1) high path, i.e., the second focus of the transfer orbit (F' from Figure A.2) is above the chord line; (2) low path, i.e., the second focus of (F'' from Figure A.2) is below the chord line.

A.2 Implementation bits

It can be demonstrated [257] the transfer time $\Delta t'$ between two positions $\vec{r}_i$ and $\vec{r}_f$ can be expressed as a function f of the semi-major axis a of the orbit connecting $\vec{r}_i$ and $\vec{r}_f$, r_i, r_f and the chord length c:

$$\Delta t' = f(a, r_i + r_f, c) \tag{A.3}$$

The aim of a generic Lambert's problem solver is thus to find the a^* of the orbit that connects $\vec{r}_i$ to $\vec{r}_f$ in the given Δt, to then reconstruct all the variables of the solution orbit (i.e., velocities $\vec{v}_i$ and $\vec{v}_f$). This corresponds to solve a non-linear equation:

$$\Delta t'(a^*) - \Delta t = f(a^*, r_i + r_f, c) - \Delta t = 0 \tag{A.4}$$

Choosing the semi-major axis a as free-parameter for solving Eq. A.4 can be inconvenient as [258]: (1) the solution is not unique, (2) the derivative of transfer time is singular when $a = a_m$ (i.e., the semi-major axis of the minimum energy ellipse). Thus, modern Lambert solvers use different independent variables to solve the time equation as in Eq. A.4. For example, Battin [38], Gooding [39] and Izzo [40] use universal variables, Avanzini [259] and Wen [260] use eccentricity, while Bate [261] and Boltz [262] use semi-latus rectum. The other ingredients that affect the performances of the Lambert solver are [258, 263]:

- The initial guess of the independent variable
- The root finder for solving Eq. A.4
- The procedure to reconstruct $\vec{v}_i$ and $\vec{v}_f$ from the independent variable

In the present work, the algorithm presented by Izzo [40] is used due to its ease of implementation and computing times [258, 263]. The full details of the algorithm are beyond the scope of the present thesis and the interested reader is referred to [40]. For the purposes of the present thesis, it suffices to say that the independent variable defined by Izzo is called ξ and it is related to the eccentric anomaly of the spacecraft on the solution orbit. The initial guess is provided by a linear approximation of the time of flight as in Eq. A.3, that is made possible thanks to the definition of the variable ξ. The Householder iteration scheme is adopted as non-linear solver to find the solution to Eq. A.4. This uses up to the second derivative of the time of flight equation with respect to the independent variable to update the initial guess towards the final solution. Velocities $\vec{v}_i$ and $\vec{v}_f$ are then found by using again the variable ξ, and relating it to the anomalies on the solutions orbit.

Appendix B

Dynamic Programming on Discrete Problems

Dynamic programming is an efficient method first introduced by Bellman [118] to guarantee global optimality in (discrete) sequential optimization problems. The underlying assumption is that the search space must be composed of discrete node such that the cost of connecting two nodes is uniquely defined (e.g., in the classical TSP variant, the cities correspond to the nodes and the cost connecting them is related to the distance that the salesman needs to travel between the two). Moreover, the problem must exhibit an optimal substructure [118], that is, the optimal solution to the problem must incorporate optimal solutions to related sub-problems, which can be solved independently [162]. Hence, the question arises of what the minimum size sub-problem is that can be solved independently. Solving independently refers here to the capacity to compute the fitness of a given part of the trajectory (e.g., distance between cities) without the need to know what the path was prior to arrive to that point. In addition, the overall problem must be formulated as a multi-stage decision process, on which the stages represent the points at which a decision is required. In multi-stage decision processes, the construction of the overall sequence of nodes is reached by making a series of lower-level choices, i.e., the selection of consecutive nodes.

At each stage, the system can be in different states, and the decision transforms the current state into another one associated to the next stage. To make optimal decisions, Bellman's principle of optimality is used. Such principle states that an optimal decision policy has the property to be independent from the history needed to arrive at the specific point. The optimal policy is thus a sequence of decisions which is the most advantageous from a pre-assigned criterion. Assuming the system to be in the state s_n at a given stage n, the optimal decision x_n^* is the one that minimizes the cost f_n resulting from the best overall policy at stages $1,\ldots,n-1$ plus a contribution c_{s_n,x_n} produced by the current decision x_n.

Thus, the optimization is carried out by using the recursive formulation:

$$\begin{cases} f_n^*(s_n) = & \min\limits_{x_n \in S_n} f_n(s_n,x_n) = f_n(s_n,x_n^*) \\ f_n^*(s_n) = & \min\limits_{x_n \in S_n} \left\{c_{s_n,x_n} + f_{n-1}^*(s_n)\right\} \end{cases} \tag{B.1}$$

where S_n is the set of admissible decisions at stage n. Therefore, at each stage of the process at which the decision is required, only the minimum path leading to the given node is saved for further consideration, which constitutes the main advantage of dynamic programming.

Translating the discussion to the multi-target space trajectory optimization, Bellman's principle of optimality would state that, regardless of the node [1] at which the spacecraft currently is, the optimal set containing this specific node would contain the optimal subset of nodes before and after the visited one (optimality refers to the cost function, typically Δv). In other words, dynamic programming answers the following question:

If the spacecraft is at a given node, what is the best sequence of nodes to come from, according to a pre-assigned fitness criterion?

[1] Recall that a node is made by a couple of planets/asteroids and their visiting epochs for the purposes

Table B.1 summarizes the variables needed for the dynamic programming transcription relevant to the MGA optimization problem. Since the whole trajectory is modelled with consecutive Lambert arcs between pair of objects in the sequence, the Δv cost for a given planet-to-planet leg depends upon the previously visited object, as discussed in Chapter 5. The Bellman's principle of optimality as from the recursive formula in Eq. B.1 is applicable only considering each state s_n as couple of planets (i.e., a node) at a given level n in the overall sequence, i.e., $n = 1, \ldots, n_{int} - 1$, where is the number of fly-by objects in the sequence. Therefore, at each stage one wants to find the overall best sub-sequence to come from, i.e., $f^*_{n-1}(s_n)$, for each state s_n that contributes with , corresponding to the Δv on the last leg attached to $f^*_{n-1}(s_n)$.

Table B.1: Summary of dynamic programming formulation for the MGA problem.

Variables	**Description**
$n = 1, \ldots, n_{int} - 1$	Number of stages, i.e., the number of fly-bys to be evaluated
s_n	Sates, i.e., planetary couples to be branched at stage n
x_n	Decision policy, i.e., selection of a given sequence of nodes up to stage n
$f^*_{n-1}(s_n)$	Optimal sequence leading to state s_n
c_{s_n,x_n}	Contribution of state s_n due to the decision policy x_n

Figure B.1 also provides a representation of dynamic programming approach to MGA trajectory optimization. Nodes are linked by means of Δv defects (see also Chapter 3). At the given node, i.e., the last one on the right, one should evaluate the best history leading to that node, i.e., $f^*_{n-1}(s_n)$, plus the contribution c_{s_n,x_n} provided by the last node. For the purposes of the present thesis, since the overall function to be minimized is the f_1 from Eq. 5.2, the contribution of the first and last node to the overall sequence should also consider infinity velocities at the departing and arrival planets. In other words, given a node at which the spacecraft currently is, only the sequence of nodes that minimize the quantity $c_{s_n,x_n} + f^*_{n-1}(s_n)$ is kept for further consideration (recall that a node is made by a couple of planets and their visiting epochs as described in Chapter 5).

of the present thesis as discussed in Chapter 5.

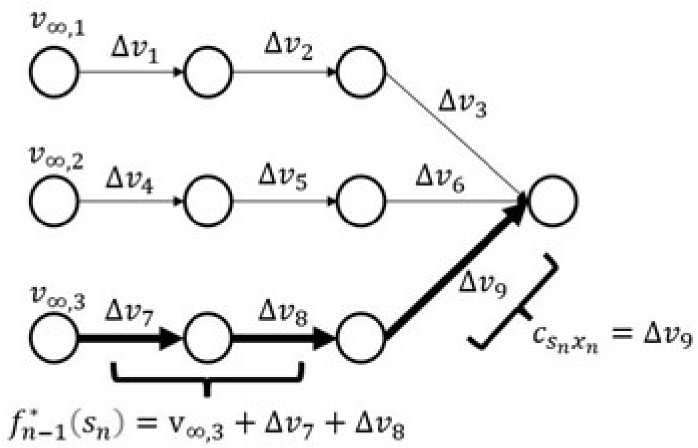


Figure B.1: Representation of dynamic programming approach to MGA trajectory optimization for a generic current node evaluated (i.e., the one at the far right). Each node represents a couple of planets and their visiting epochs. Connection between two nodes is given by Δv defects. The contribution of first nodes is explicitly reported as infinity velocity needed to leave the first planet, i.e., v_∞. Bold arrows highlight the optimal path leading to the current node under consideration.

Appendix C

Toolboxes

In this Appendix, the functioning of the produced toolboxes is discussed with practical examples. The toolboxes are encoded in MATLAB. The following sections describe AUTOMATE (section C.1), ASTRA (section C.2), RESTOUR (section C.3) and DYNAMIS (section C.4). The toolboxes can be found at [74]

C.1 AUTOMATE

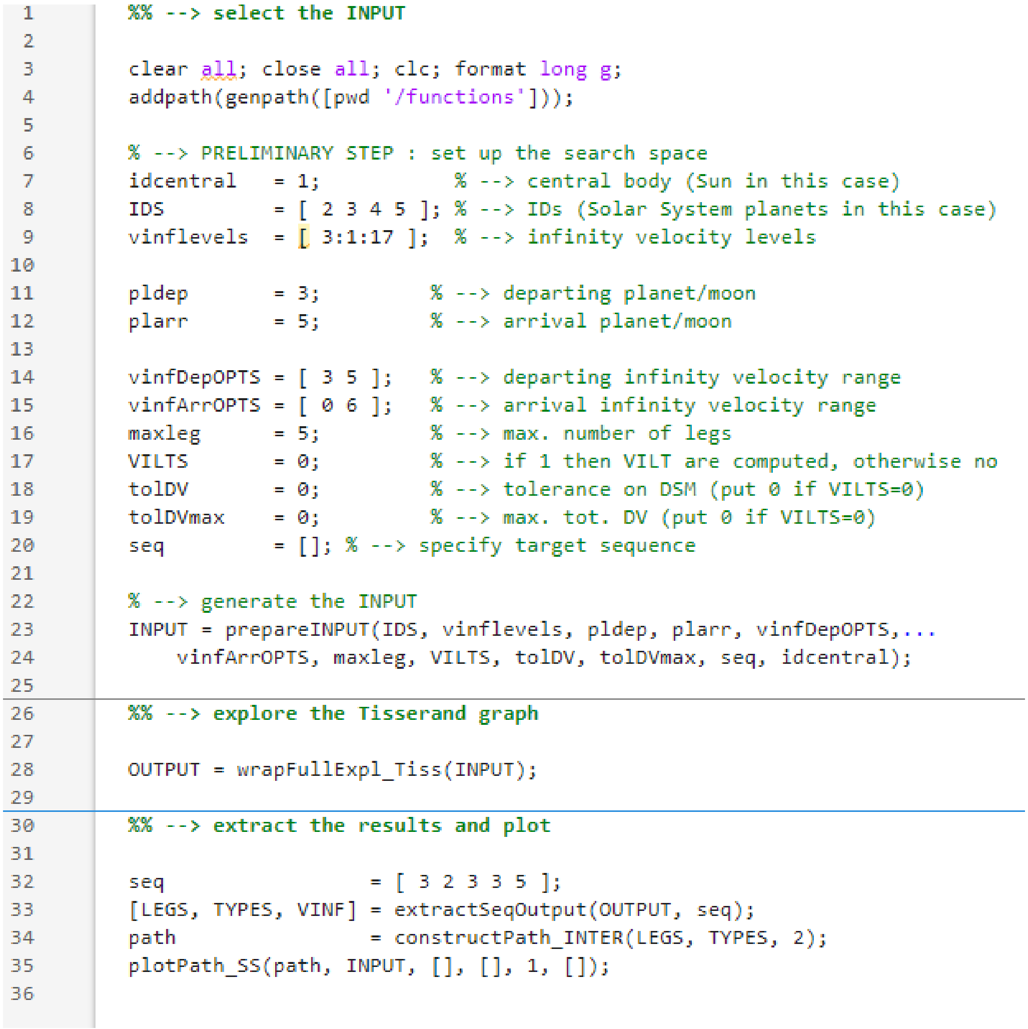
AUTOmatic Multiple-gravity Assist with Tisserand Exploration (AUTOMATE) allows to automatically explore Tisserand graphs, and thus to list all the possible planetary sequences and DSMs that reach specific orbital regions. AUTOMATE main engine is presented in Algorithm 3.

AUTOMATE allows to search for fly-by tours within three different regimes:

- the Solar System, the main attracting body being the Sun and fly-by objects being the planets (ID: 1.Mercury, 2.Venus, 3.Earth, 4.Mars, 5.Jupiter, 6.Saturn, 7.Uranus, 8.Neptune)
- the Jupiter Moon System, the main attracting body being Jupiter and the fly-by objects being its biggest moons (ID: 1.Io, 2.Europa, 3.Ganymede, 4.Callisto)

- the Saturn Moon System, the main attracting body being Saturn and the fly-by objects being its biggest moons (ID: 1.Enceladus, 2.Tethys, 3.Dione, 4.Rhea, 5.Titan)

A typical MATLAB script to call AUTOMATE starts as in Figure C.1. One first needs to ensure that the folder named 'functions' is within the path (line 4). Then, one of the orbital regimes should be selected, by setting the appropriate central body (ID: 1. Sun, 5.Jupiter, 6.Saturn), the fly-by bodies and the infinity velocity contours (lines 6-9).

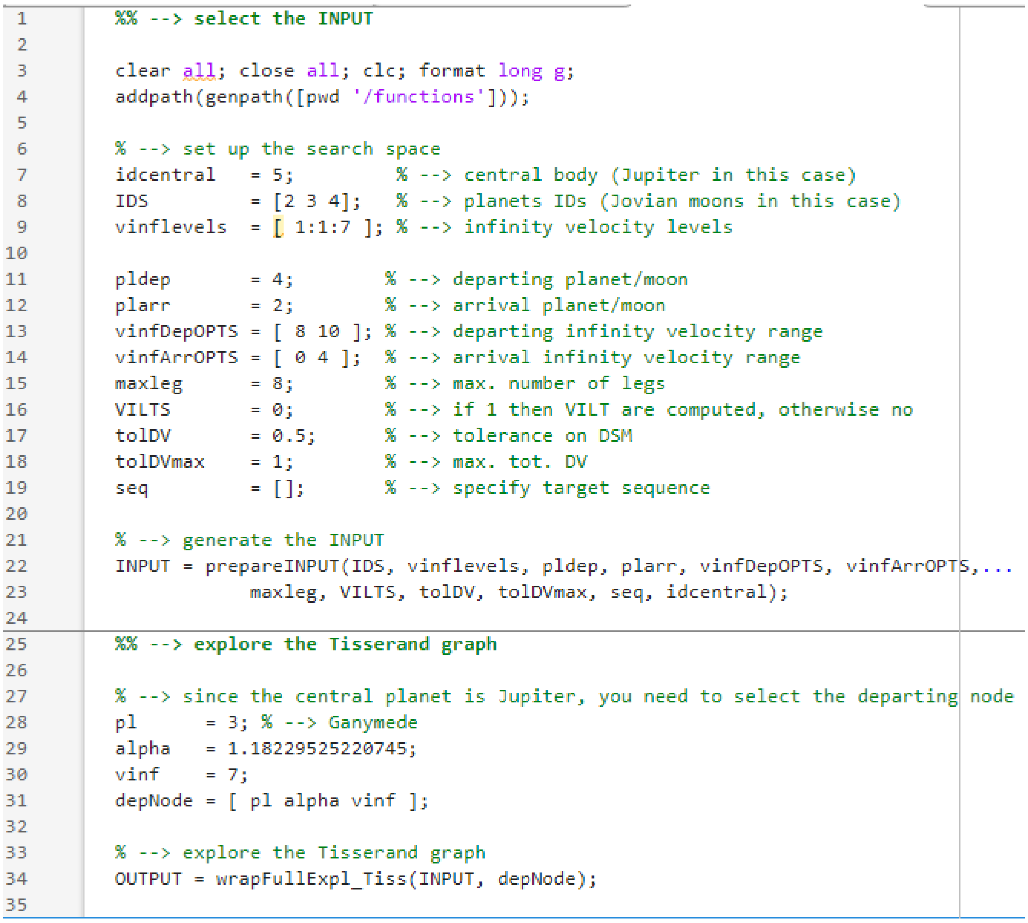


```
1   %% --> select the INPUT
2
3   clear all; close all; clc; format long g;
4   addpath(genpath([pwd '/functions']));
5
6   % --> PRELIMINARY STEP : set up the search space
7   idcentral    = 1;           % --> central body (Sun in this case)
8   IDS          = [ 2 3 4 5 ]; % --> IDs (Solar System planets in this case)
9   vinflevels   = [ 3:1:17 ];  % --> infinity velocity levels
10
11  pldep        = 3;         % --> departing planet/moon
12  plarr        = 5;         % --> arrival planet/moon
13
14  vinfDepOPTS = [ 3 5 ];    % --> departing infinity velocity range
15  vinfArrOPTS = [ 0 6 ];    % --> arrival infinity velocity range
16  maxleg       = 5;         % --> max. number of legs
17  VILTS        = 0;         % --> if 1 then VILT are computed, otherwise no
18  tolDV        = 0;         % --> tolerance on DSM (put 0 if VILTS=0)
19  tolDVmax     = 0;         % --> max. tot. DV (put 0 if VILTS=0)
20  seq          = []; % --> specify target sequence
21
22  % --> generate the INPUT
23  INPUT = prepareINPUT(IDS, vinflevels, pldep, plarr, vinfDepOPTS,...
24      vinfArrOPTS, maxleg, VILTS, tolDV, tolDVmax, seq, idcentral);
25
26  %% --> explore the Tisserand graph
27
28  OUTPUT = wrapFullExpl_Tiss(INPUT);
29
30  %% --> extract the results and plot
31
32  seq                 = [ 3 2 3 3 5 ];
33  [LEGS, TYPES, VINF] = extractSeqOutput(OUTPUT, seq);
34  path                = constructPath_INTER(LEGS, TYPES, 2);
35  plotPath_SS(path, INPUT, [], [], 1, []);
36
```

Figure C.1: AUTOMATE script for Solar System exploration.

One should then select the departing and arrival planets (lines 11-12). Maximum and minimum infinity velocity at departure and arrival are also requested, as well as the maximum number of legs (lines 14-16). Some optional input follow. In particular, one can set $VILTS = 0$ or $VILTS = 1$ (line 17), if VILTs are to be used or not, respectively (see also

Algorithm 2). If $VILTS = 1$, then one can select the maximum Δv admissible on each MGA leg and the maximum Δv for the overall MGA sequence (lines 18-19). Additionally, if a specific sequence is searched on the Tisserand map, one can explicitly specify it (line 20); otherwise, one simply sets $seq = []$. Input are then collected in a structure named $INPUT$ (lines 22-24). The function 'prepareINPUT.m' computes and stores all the possible intersections on Tisserand map (see Algorithm 1), as well as VILTs transfers (2), if $VILTS = 1$.

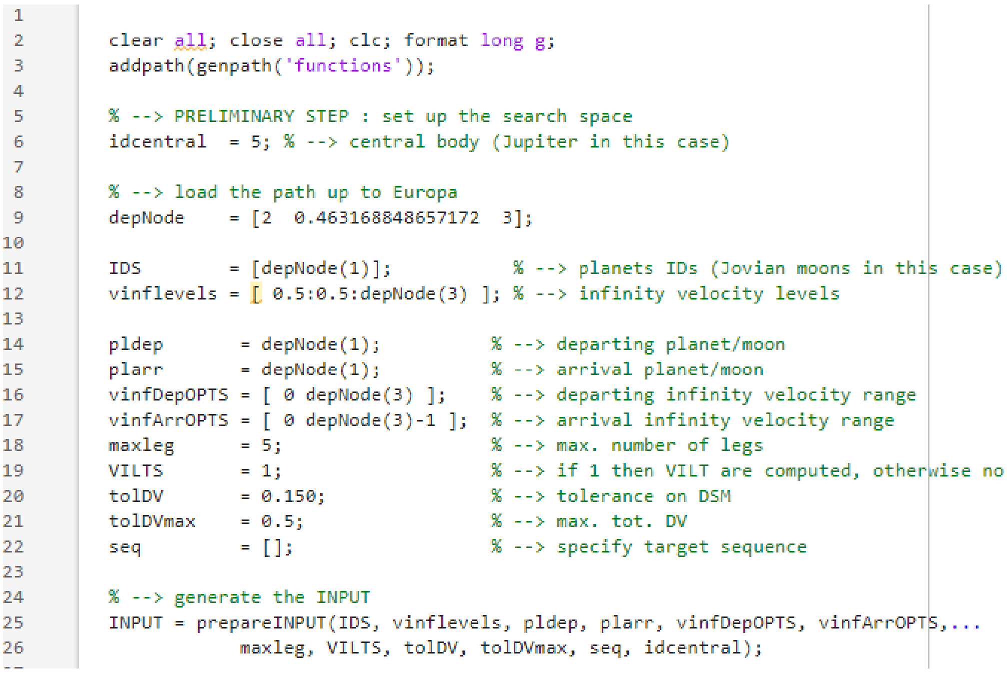


```
1    %% --> select the INPUT
2
3    clear all; close all; clc; format long g;
4    addpath(genpath([pwd '/functions']));
5
6    % --> set up the search space
7    idcentral    = 5;          % --> central body (Jupiter in this case)
8    IDS          = [2 3 4];    % --> planets IDs (Jovian moons in this case)
9    vinflevels   = [ 1:1:7 ]; % --> infinity velocity levels
10
11   pldep        = 4;         % --> departing planet/moon
12   plarr        = 2;         % --> arrival planet/moon
13   vinfDepOPTS = [ 8 10 ]; % --> departing infinity velocity range
14   vinfArrOPTS = [ 0 4 ];  % --> arrival infinity velocity range
15   maxleg       = 8;         % --> max. number of legs
16   VILTS        = 0;         % --> if 1 then VILT are computed, otherwise no
17   tolDV        = 0.5;       % --> tolerance on DSM
18   tolDVmax     = 1;         % --> max. tot. DV
19   seq          = [];        % --> specify target sequence
20
21   % --> generate the INPUT
22   INPUT = prepareINPUT(IDS, vinflevels, pldep, plarr, vinfDepOPTS, vinfArrOPTS,...
23               maxleg, VILTS, tolDV, tolDVmax, seq, idcentral);
24
25   %% --> explore the Tisserand graph
26
27   % --> since the central planet is Jupiter, you need to select the departing node
28   pl      = 3; % --> Ganymede
29   alpha   = 1.18229525220745;
30   vinf    = 7;
31   depNode = [ pl alpha vinf ];
32
33   % --> explore the Tisserand graph
34   OUTPUT = wrapFullExpl_Tiss(INPUT, depNode);
35
```

Figure C.2: AUTOMATE script for Jupiter Moon System exploration.

The exploration is then performed (line 28) and the results are saved in a structure named $OUTPUT$. Legs and types of transfers (i.e., resonances, VILTs, intersections) are saved in $OUTPUT.LEGS$ and $OUTPUT.TYPES$. For representation purposes, one can specify the sequence to plot on the Tisserand map (line 32). The code extracts the sequence from $OUTPUT$ structure (line 33) and then constructs the path (line 34). The function *plotPath_SS.m* is then used for representing the sequence on Tisserand map (many options exist for plotting, like the colour of the path, the colour of the Tisserand contours, legend, and so on). The function 'extractSeqOutput.m' (line 33) also provides information on minimum and maximum infinity velocity at each planetary encounter (variable *VINF*).

If one wants to construct more complex MGA sequences, e.g. within the Jupiter or Saturn system, a script like the one in Figure C.2 should be used. From Figure C.2, similarly to Figure C.1, one selects the input (lines 6-23) that are relevant for the scenario under consideration. Since the central body is Jupiter, AUTOMATE requires a departing node on the Tisserand map (lines 27-31). This can be the orbit reached by the spacecraft after the injection within the planet SOI (the same applies when Saturn is the central body). The exploration can then be performed (line 34). The specific setup shown in Figure C.2 can be used to obtain Figure 4.10.a.



```
1
2   clear all; close all; clc; format long g;
3   addpath(genpath('functions'));
4
5   % --> PRELIMINARY STEP : set up the search space
6   idcentral  = 5; % --> central body (Jupiter in this case)
7
8   % --> load the path up to Europa
9   depNode    = [2  0.463168848657172  3];
10
11  IDS        = [depNode(1)];            % --> planets IDs (Jovian moons in this case)
12  vinflevels = [ 0.5:0.5:depNode(3) ]; % --> infinity velocity levels
13
14  pldep       = depNode(1);           % --> departing planet/moon
15  plarr       = depNode(1);           % --> arrival planet/moon
16  vinfDepOPTS = [ 0 depNode(3) ];     % --> departing infinity velocity range
17  vinfArrOPTS = [ 0 depNode(3)-1 ];   % --> arrival infinity velocity range
18  maxleg      = 5;                    % --> max. number of legs
19  VILTS       = 1;                    % --> if 1 then VILT are computed, otherwise no
20  tolDV       = 0.150;                % --> tolerance on DSM
21  tolDVmax    = 0.5;                  % --> max. tot. DV
22  seq         = [];                   % --> specify target sequence
23
24  % --> generate the INPUT
25  INPUT = prepareINPUT(IDS, vinflevels, pldep, plarr, vinfDepOPTS, vinfArrOPTS,...
26              maxleg, VILTS, tolDV, tolDVmax, seq, idcentral);
```

Figure C.3: AUTOMATE script for Europa leveraging options.

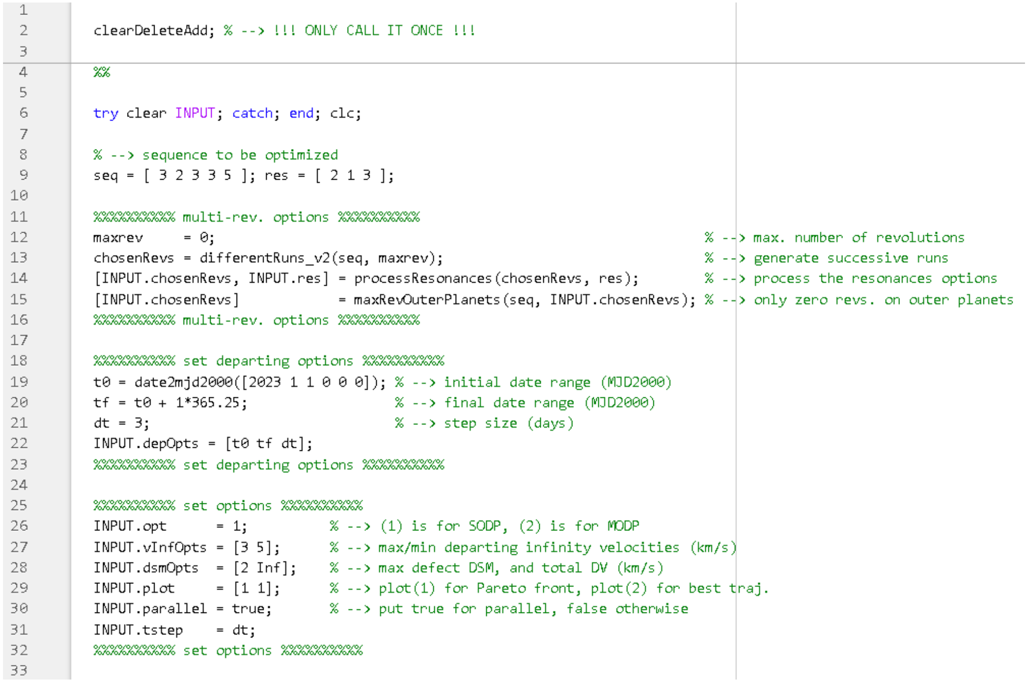
To obtain Figure 4.10.b, i.e., leveraging sequence at Europa, one can explicitly look for VILTs transfers with a single moon (it also works with Solar System planets). Figure C.3 shows an example. As before, one should specify the departing node (lines 8-9), that can be an intersection with a previously-visited moon. VILTs option is now set to 1 (line 19), to allow the search for VILTs transfer and thus to reduce the infinity velocity progressively. The exploration can be then performed by launching the same command (line 34) as in Figure C.2.

C.2 ASTRA

Automatic Swing-by TRAjectories (ASTRA) implements SODP and MODP to look for departing dates and transfer times for any given sequence of fly-by bodies (e.g., provided by the Tisserand exploration from AUTOMATE).

A MATLAB script for setting ASTRA input is represented in Figure C.4. The script allows to optimize an EVEEJ sequence with a 2:1 resonant transfer on the third leg.

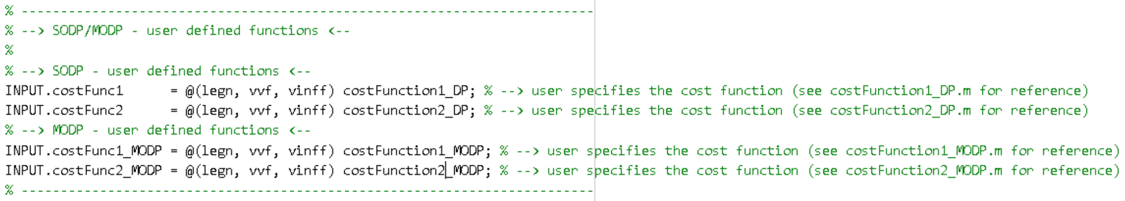
Line 2 calls a script (’*clearDeleteAdd.m*’) that is useful to add ASTRA functions to the working path and to build C++ based mex files like the multi-revolutions Lambert solver and the function to compute defects.



```
1
2   clearDeleteAdd; % --> !!! ONLY CALL IT ONCE !!!
3
4   %%
5
6   try clear INPUT; catch; end; clc;
7
8   % --> sequence to be optimized
9   seq = [ 3 2 3 3 5 ]; res = [ 2 1 3 ];
10
11  %%%%%%%%%% multi-rev. options %%%%%%%%%%
12  maxrev     = 0;                                                        % --> max. number of revolutions
13  chosenRevs = differentRuns_v2(seq, maxrev);                            % --> generate successive runs
14  [INPUT.chosenRevs, INPUT.res] = processResonances(chosenRevs, res);    % --> process the resonances options
15  [INPUT.chosenRevs]            = maxRevOuterPlanets(seq, INPUT.chosenRevs); % --> only zero revs. on outer planets
16  %%%%%%%%%% multi-rev. options %%%%%%%%%%
17
18  %%%%%%%%%% set departing options %%%%%%%%%%
19  t0 = date2mjd2000([2023 1 1 0 0 0]); % --> initial date range (MJD2000)
20  tf = t0 + 1*365.25;                  % --> final date range (MJD2000)
21  dt = 3;                              % --> step size (days)
22  INPUT.depOpts = [t0 tf dt];
23  %%%%%%%%%% set departing options %%%%%%%%%%
24
25  %%%%%%%%%% set options %%%%%%%%%%
26  INPUT.opt      = 1;          % --> (1) is for SODP, (2) is for MODP
27  INPUT.vInfOpts = [3 5];      % --> max/min departing infinity velocities (km/s)
28  INPUT.dsmOpts  = [2 Inf];    % --> max defect DSM, and total DV (km/s)
29  INPUT.plot     = [1 1];      % --> plot(1) for Pareto front, plot(2) for best traj.
30  INPUT.parallel = true;       % --> put true for parallel, false otherwise
31  INPUT.tstep    = dt;
32  %%%%%%%%%% set options %%%%%%%%%%
33
```

Figure C.4: Script for ASTRA input.

One needs to specify the sequence of planets to be optimized alongside specific resonant legs (line 9). If a resonant transfer is present on a planet-to-planet leg, one specifies it as $res = \begin{bmatrix} 2 & 1 & 3 \end{bmatrix}$, meaning that the 2:1 resonance is present on the 3rd leg. If no resonances are present in the transfer, one simply sets $res = []$. One then needs to select the maximum number of revolutions per leg (line 12). All the possible permutations of revolutions per leg are produced (lines 13-14) and those options that have more than one revolution on outer planets (i.e., Jupiter, Saturn, Uranus, Neptune) are removed (line 15) [1].



```
% ------------------------------------------------------------------------
% --> SODP/MODP - user defined functions <--
%
% --> SODP - user defined functions <--
INPUT.costFunc1      = @(legn, vvf, vinff) costFunction1_DP; % --> user specifies the cost function (see costFunction1_DP.m for reference)
INPUT.costFunc2      = @(legn, vvf, vinff) costFunction2_DP; % --> user specifies the cost function (see costFunction2_DP.m for reference)
% --> MODP - user defined functions <--
INPUT.costFunc1_MODP = @(legn, vvf, vinff) costFunction1_MODP; % --> user specifies the cost function (see costFunction1_MODP.m for reference)
INPUT.costFunc2_MODP = @(legn, vvf, vinff) costFunction2_MODP; % --> user specifies the cost function (see costFunction2_MODP.m for reference)
% ------------------------------------------------------------------------
```

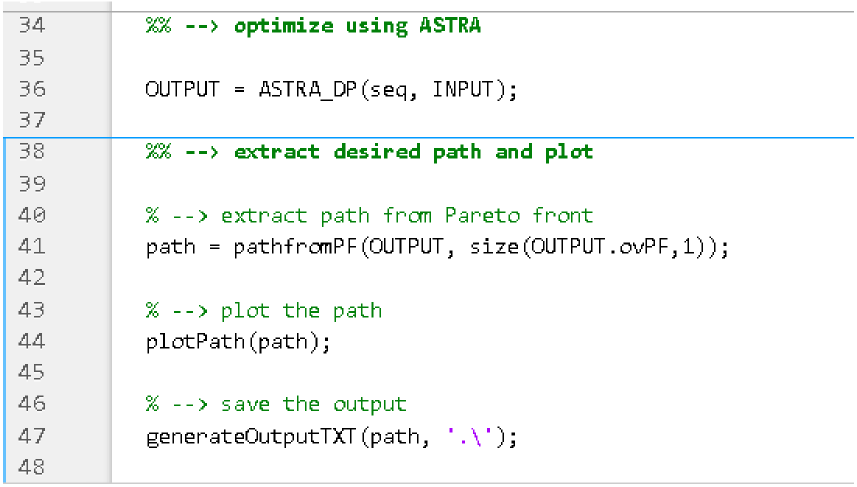
Figure C.5: Script for specifying objective functions in ASTRA.

Departing dates range should then be defined (lines 18-23). Additional options should then be defined. One selects: (1) the type of optimization to be performed, either SODP or MODP (line 26); (2) the boundaries on departure v_∞ (line 17), on the maximum Δv per leg and on the maximum accumulated Δv (line 18); (3) plot options (line 29); (4) if parallel computing is to be used (line 30); (5) the grid step sizes (line 31) [2].

Without specifying any other input, ASTRA automatically sets the bounds for planet-to-planet transfer times. If Mercury, Venus and Earth are the only planets involved in the leg, transfer times are bounded as $T \in [50, 750]$ days. If Mars is in the leg, then $T \in [50, 850]$ days. If outer planets are present in the leg (i.e., Jupiter, Saturn, Uranus, Neptune), then $T \in [5002500]$ days. If multiple revolutions are admitted in the leg, then $T \in [400, 2500]$ days. The user can also set up their own bounds by adding a line like $INPUT.TOF\_LIM = \begin{bmatrix} \begin{bmatrix} 50 & 400 \end{bmatrix}; & \begin{bmatrix} 50 & 400 \end{bmatrix}; & \begin{bmatrix} 50 & 400 \end{bmatrix}; & \begin{bmatrix} 500 & 1000 \end{bmatrix} \end{bmatrix}$ (units are days).

[1] This is optional.

[2] For outer planets (i.e., Jupiter, Saturn, Uranus, Neptune), the step size is automatically doubled.



```
34    %% --> optimize using ASTRA
35
36    OUTPUT = ASTRA_DP(seq, INPUT);
37
38    %% --> extract desired path and plot
39
40    % --> extract path from Pareto front
41    path = pathfromPF(OUTPUT, size(OUTPUT.ovPF,1));
42
43    % --> plot the path
44    plotPath(path);
45
46    % --> save the output
47    generateOutputTXT(path, '.\');
48
```

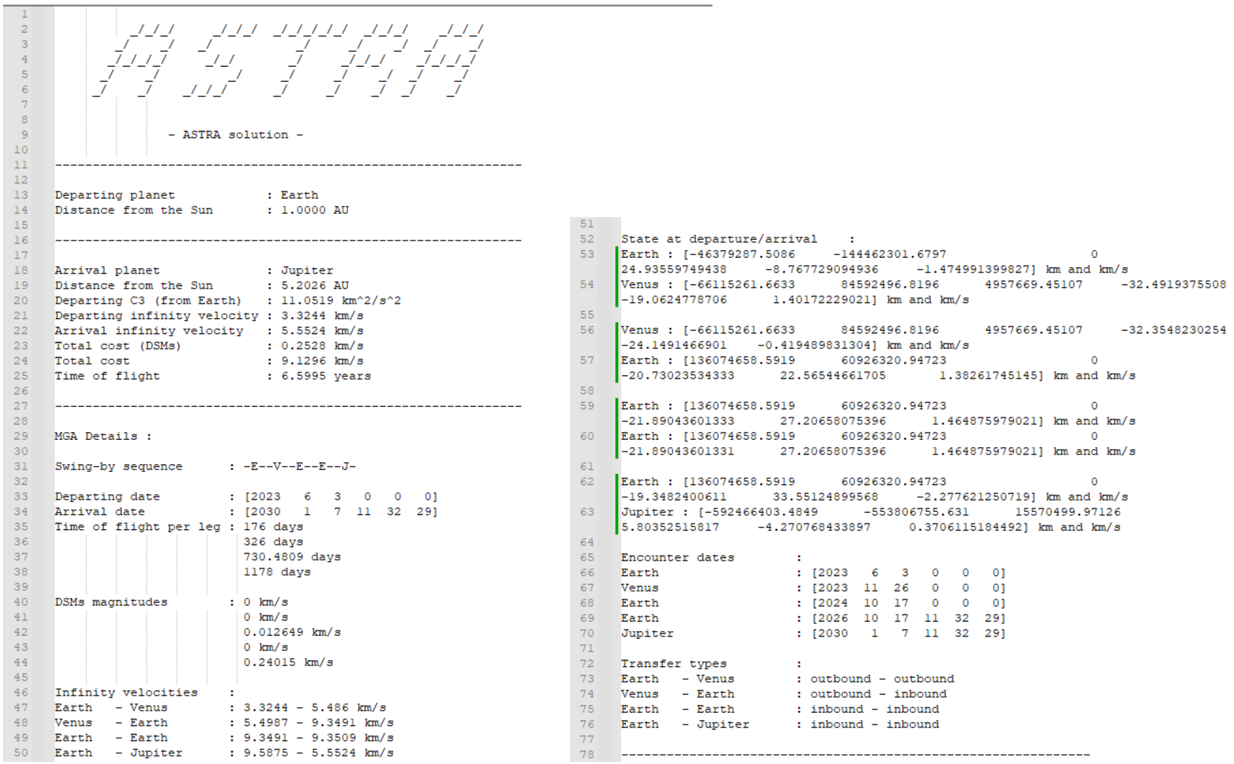
Figure C.6: Script for launching ASTRA and post-processing.

By default, if SODP is chosen, ASTRA minimizes the first of Eq. 5.2, otherwise, also the overall transfer time is minimized. The user can specify their own objectives by defining MATLAB anonymous functions as in Figure C.5. By default, ASTRA does not apply any bounds on the infinity velocities at planetary encounters. The user can specify their own bounds (e.g., informed by a Tisserand-based exploration as from section C.1) by adding a line like $INPUT.VINF\_LIM = \begin{bmatrix}\begin{bmatrix}3 & 5\end{bmatrix}; & \begin{bmatrix}5 & 10\end{bmatrix}; & \begin{bmatrix}5 & 10\end{bmatrix}; & \begin{bmatrix}3 & 6\end{bmatrix}\end{bmatrix}$ (units are km/s).

ASTRA can then be run as in Figure C.6. Results are saved in a structure called $OUTPUT$ (line 36). To extract information from it, one can look at the field $OUTPUT.ovPF$ where the Pareto front is saved (objectives are organized in columns [3]). The user then selects the row number of $OUTPUT.PF$ for the path to display and calls the function $pathfromPF.m$ to build the path (line 41). One can then plot it with the function $plotPath.m$ (line 44) and export its values as a .txt file using $generateOutputTXT.m$ (line 47) [4]. A typical ASTRA .txt output file is shown in Figure C.7.

[3] Last column encodes an identifier for the position of the specific path in the final results set.

[4] This function requires in input the path and the folder where to save the .txt file



```

        _/_/_/      _/_/_/  _/_/_/_/_/  _/_/_/      _/_/_/
     _/    _/    _/            _/      _/    _/  _/    _/
    _/_/_/_/      _/_/        _/      _/_/_/    _/_/_/_/
   _/    _/          _/      _/      _/    _/  _/    _/
  _/    _/    _/_/_/        _/      _/    _/  _/    _/


              - ASTRA solution -

-------------------------------------------------------------

Departing planet              : Earth
Distance from the Sun         : 1.0000 AU

-------------------------------------------------------------

Arrival planet                : Jupiter
Distance from the Sun         : 5.2026 AU
Departing C3 (from Earth)     : 11.0519 km^2/s^2
Departing infinity velocity : 3.3244 km/s
Arrival infinity velocity     : 5.5524 km/s
Total cost (DSMs)             : 0.2528 km/s
Total cost                    : 9.1296 km/s
Time of flight                : 6.5995 years

-------------------------------------------------------------

MGA Details :

Swing-by sequence      : -E--V--E--E--J-

Departing date         : [2023   6   3   0   0   0]
Arrival date           : [2030   1   7  11  32  29]
Time of flight per leg : 176 days
                         326 days
                         730.4809 days
                         1178 days

DSMs magnitudes        : 0 km/s
                         0 km/s
                         0.012649 km/s
                         0 km/s
                         0.24015 km/s

Infinity velocities    :
Earth   - Venus        : 3.3244 - 5.486 km/s
Venus   - Earth        : 5.4987 - 9.3491 km/s
Earth   - Earth        : 9.3491 - 9.3509 km/s
Earth   - Jupiter      : 9.5875 - 5.5524 km/s
```

(a)

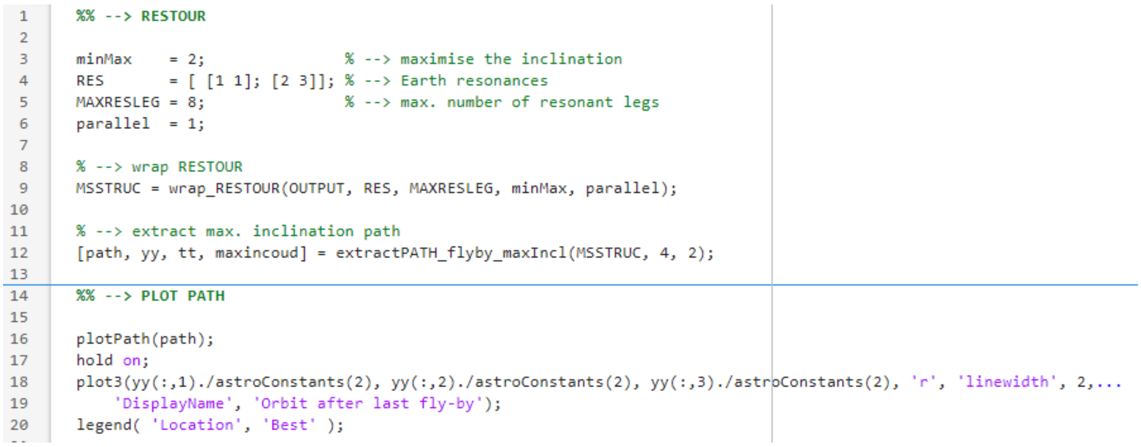
```

State at departure/arrival     :
Earth : [-46379287.5086     -144462301.6797                   0
24.93559749438     -8.767729094936     -1.474991399827] km and km/s
Venus : [-66115261.6633      84592496.8196      4957669.45107     -32.4919375508
-19.0624778706      1.40172229021] km and km/s

Venus : [-66115261.6633      84592496.8196      4957669.45107     -32.3548230254
-24.1491466901    -0.419489831304] km and km/s
Earth : [136074658.5919      60926320.94723                   0
-20.73023534333      22.56544661705       1.38261745145] km and km/s

Earth : [136074658.5919      60926320.94723                   0
-21.89043601333      27.20658075396      1.464875979021] km and km/s
Earth : [136074658.5919      60926320.94723                   0
-21.89043601331      27.20658075396      1.464875979021] km and km/s

Earth : [136074658.5919      60926320.94723                   0
-19.3482400611      33.55124899568     -2.277621250719] km and km/s
Jupiter : [-592466403.4849      -553806755.631      15570499.97126
5.80352515817     -4.270768433897     0.3706115184492] km and km/s

Encounter dates             :
Earth                       : [2023   6   3   0   0   0]
Venus                       : [2023  11  26   0   0   0]
Earth                       : [2024  10  17   0   0   0]
Earth                       : [2026  10  17  11  32  29]
Jupiter                     : [2030   1   7  11  32  29]

Transfer types              :
Earth   - Venus             : outbound - outbound
Venus   - Earth             : outbound - inbound
Earth   - Earth             : inbound - inbound
Earth   - Jupiter           : inbound - inbound

-------------------------------------------------------------
```

(b)

Figure C.7: ASTRA output file example.

C.3 RESTOUR

Resonant TOUR (RESTOUR) uses $OUTPUT$ structure from ASTRA (as from section C.2) to find maximum inclination orbits exploiting successive resonant fly-bys with the last planet of the sequence optimized by ASTRA.



```matlab
%% --> RESTOUR

minMax    = 2;                % --> maximise the inclination
RES       = [ [1 1]; [2 3]]; % --> Earth resonances
MAXRESLEG = 8;                % --> max. number of resonant legs
parallel  = 1;

% --> wrap RESTOUR
MSSTRUC = wrap_RESTOUR(OUTPUT, RES, MAXRESLEG, minMax, parallel);

% --> extract max. inclination path
[path, yy, tt, maxincoud] = extractPATH_flyby_maxIncl(MSSTRUC, 4, 2);

%% --> PLOT PATH

plotPath(path);
hold on;
plot3(yy(:,1)./astroConstants(2), yy(:,2)./astroConstants(2), yy(:,3)./astroConstants(2), 'r', 'linewidth', 2,...
    'DisplayName', 'Orbit after last fly-by');
legend( 'Location', 'Best' );
```

Figure C.8: Script for using RESTOUR.

Figure C.8 shows how to use RESTOUR. In particular, one needs to select if the inclination should be maximised or not by setting or , respectively (line 3). The available resonances at the given planet should be specified (line 4), as well as the maximum number of resonant legs (line 5), and the option to use parallel computing (line 6). RESTOUR can then be used (line9). As said, the structure $OUTPUT$ from ASTRA (see section C.2) should be used in input.

Additionally, if the spacecraft needs to perform the fly-by with the last planet of the resonant tour, one can extract the desired path and then perform a fly-by (i.e., finding (r_p, γ)) to maximise inclination (line 12). Lines 16-20 are for plotting. The presented script allows for reproducing the trajectory shown in Figure 4.17.

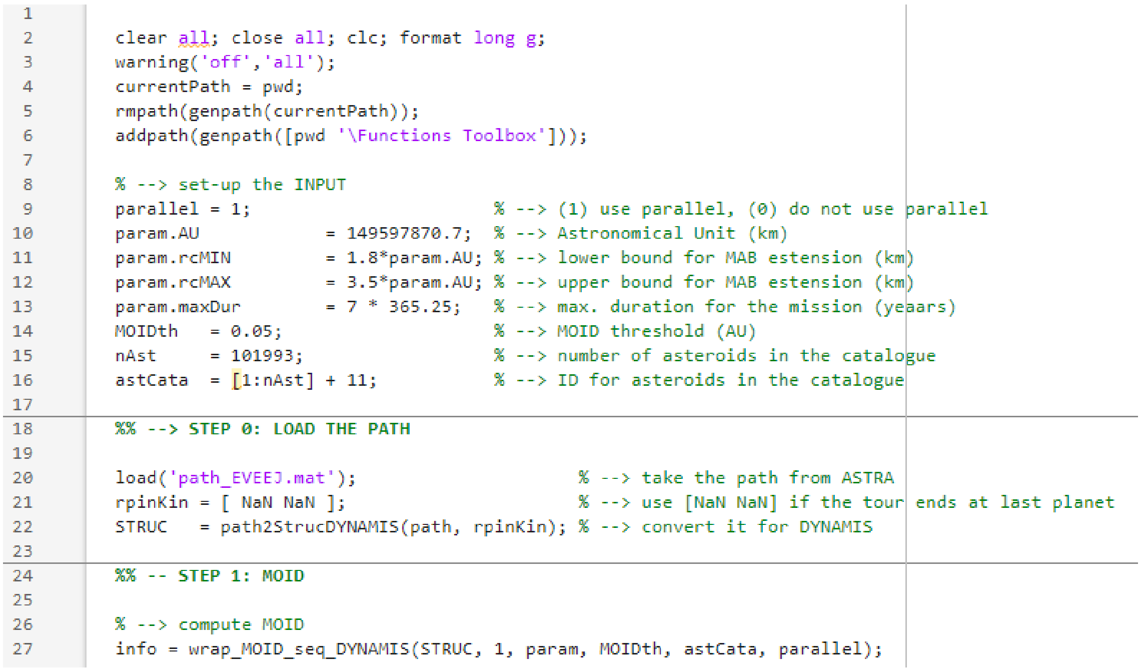
C.4 DYNAMIS

DYNamic programming for Asteroids Missions (DYNAMIS) needs input from ASTRA to look for asteroids that are close to a reference MGA trajectory. Figure C.9 shows the input needed by DYNAMIS. Particularly, one needs to select the maximum duration of the mission (line 13) [5] and the d_{thr} for MOID pruning (line 14), as from Chapter 6.

An MGA path is loaded (line 20), e.g., as coming from ASTRA output (see again section C.2), and it is processed to produce the variable $STRUC$ (line 22) via the function '$path2StrucDYNAMIS$'. In the example provided, an EVEEJ sequence is loaded, departing in October 2037. This function requires as an additional input a vector $rpinKin$ encoding (r_p, γ), i.e., the fly-by parameters for the last planet in the sequence. If the transfer ends at such planet, without a fly-by, one simply sets $rpinKin = [NaN, NaN]$. The MOID search can then be initiated (line 27). This produces a structure called $info$, where all the asteroids that are close-by the selected path are saved.

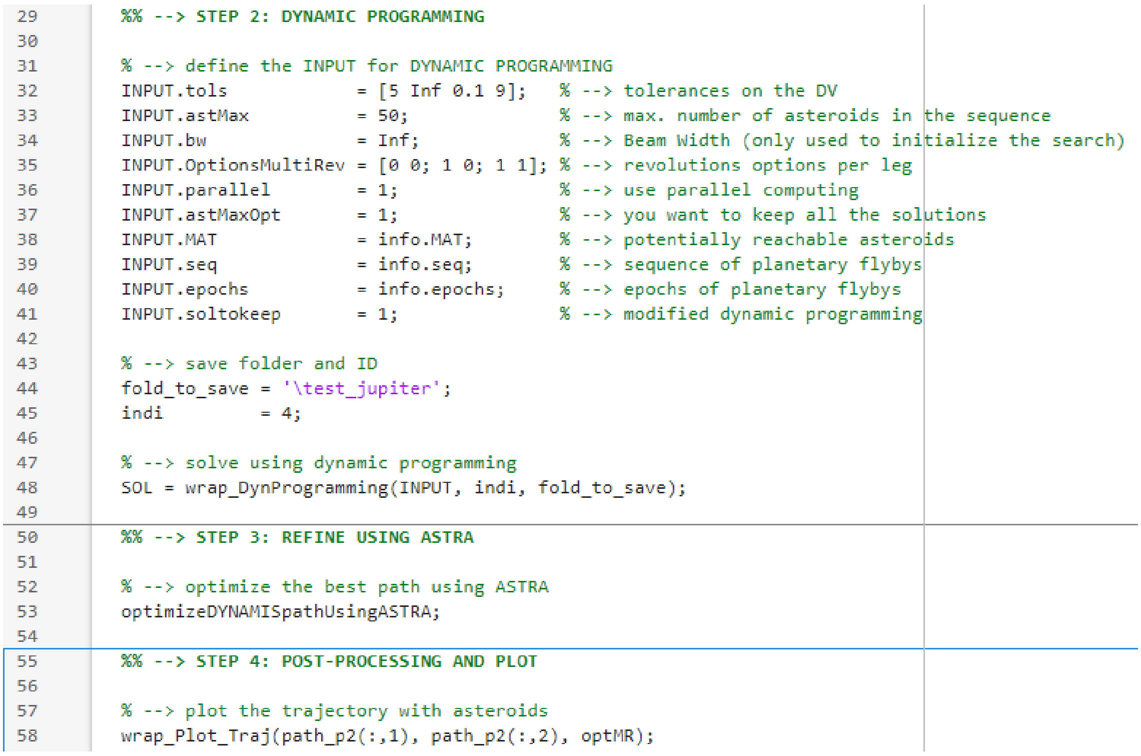
After the MOID search, dynamic programming can be used to look for asteroids se-

[5] The maximum duration of the mission is relevant only if the spacecraft performs a fly-by with the last planet in the MGA sequence. Otherwise, the maximum duration is assumed to be the time needed to reach



```
1
2   clear all; close all; clc; format long g;
3   warning('off','all');
4   currentPath = pwd;
5   rmpath(genpath(currentPath));
6   addpath(genpath([pwd '\Functions Toolbox']));
7
8   % --> set-up the INPUT
9   parallel = 1;                       % --> (1) use parallel, (0) do not use parallel
10  param.AU            = 149597870.7;  % --> Astronomical Unit (km)
11  param.rcMIN         = 1.8*param.AU; % --> lower bound for MAB estension (km)
12  param.rcMAX         = 3.5*param.AU; % --> upper bound for MAB estension (km)
13  param.maxDur        = 7 * 365.25;   % --> max. duration for the mission (yeaars)
14  MOIDth    = 0.05;                   % --> MOID threshold (AU)
15  nAst      = 101993;                 % --> number of asteroids in the catalogue
16  astCata   = [1:nAst] + 11;          % --> ID for asteroids in the catalogue
17
18  %% --> STEP 0: LOAD THE PATH
19
20  load('path_EVEEJ.mat');                     % --> take the path from ASTRA
21  rpinKin = [ NaN NaN ];                      % --> use [NaN NaN] if the tour ends at last planet
22  STRUC   = path2StrucDYNAMIS(path, rpinKin); % --> convert it for DYNAMIS
23
24  %% -- STEP 1: MOID
25
26  % --> compute MOID
27  info = wrap_MOID_seq_DYNAMIS(STRUC, 1, param, MOIDth, astCata, parallel);
```

Figure C.9: Script for ASTRA input.



```
29  %% --> STEP 2: DYNAMIC PROGRAMMING
30
31  % --> define the INPUT for DYNAMIC PROGRAMMING
32  INPUT.tols             = [5 Inf 0.1 9];   % --> tolerances on the DV
33  INPUT.astMax           = 50;              % --> max. number of asteroids in the sequence
34  INPUT.bw               = Inf;             % --> Beam Width (only used to initialize the search)
35  INPUT.OptionsMultiRev  = [0 0; 1 0; 1 1]; % --> revolutions options per leg
36  INPUT.parallel         = 1;               % --> use parallel computing
37  INPUT.astMaxOpt        = 1;               % --> you want to keep all the solutions
38  INPUT.MAT              = info.MAT;        % --> potentially reachable asteroids
39  INPUT.seq              = info.seq;        % --> sequence of planetary flybys
40  INPUT.epochs           = info.epochs;     % --> epochs of planetary flybys
41  INPUT.soltokeep        = 1;               % --> modified dynamic programming
42
43  % --> save folder and ID
44  fold_to_save = '\test_jupiter';
45  indi         = 4;
46
47  % --> solve using dynamic programming
48  SOL = wrap_DynProgramming(INPUT, indi, fold_to_save);
49
50  %% --> STEP 3: REFINE USING ASTRA
51
52  % --> optimize the best path using ASTRA
53  optimizeDYNAMISpathUsingASTRA;
54
55  %% --> STEP 4: POST-PROCESSING AND PLOT
56
57  % --> plot the trajectory with asteroids
58  wrap_Plot_Traj(path_p2(:,1), path_p2(:,2), optMR);
```

Figure C.10: Script for calling dynamic programming using DYNAMIS.

quences. Figure C.10 shows the remaining steps. One needs to select pruning criteria via the variable $INPUT.tols$ (line 32). In particular, $INPUT.tols(1)$ is for the maximum

the last planet.

$v_{\infty,dep}$; $INPUT.tols(2)$ is for the maximum Δv intended as the sum of all the DSMs in the transfer; $INPUT.tols(3)$ is for the object-to-object maximum DSM; $INPUT.tols(4)$ is for the maximum Δv intended as the sum of all the DSMs and $v_{\infty,dep}$. One can then select the number of asteroids in the sequence (line 33). If the user wants to maximise the number of asteroids in the sequence, $INPUT.astMax = 1e99$ should be used. The user is also free to select the parameter N for the modified dynamic programming approach (see Chapter 6). This can be done by choosing the parameter $INPUT.soltokeep$ (line 41). One should recall that having $N = 1$ corresponds to a SODP search. Lines 43-45 are for saving purposes. The search using dynamic programming can then be run (line 48). The results are saved in the structure SOL.

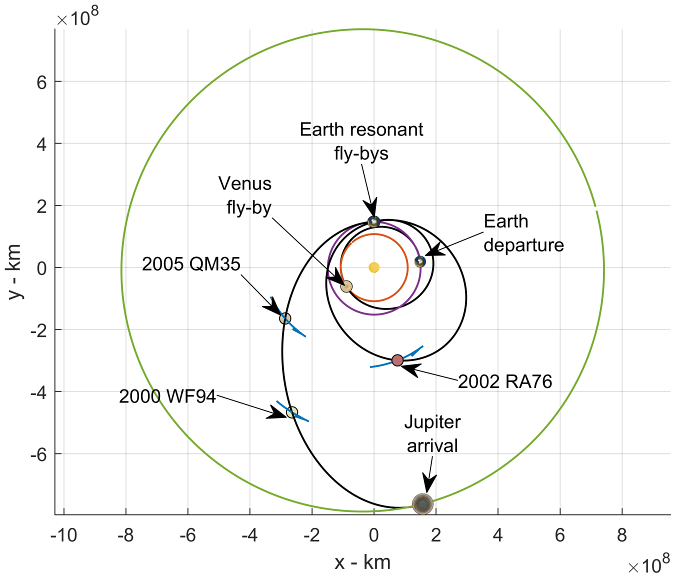


Figure C.11: EVEEJ trajectory visiting 3 asteroids.

Additionally, line 53 uses a script called '$optimizedDYNAMISpathUsingASTRA.m$' that takes the optimal path from SOL and further refines it using ASTRA, by adjusting the departing dates and transfer times. One notices that this already gives very accurate results with the advantage of ASTRA computational efficiency, even without explicitly looking for actual DSMs. This is because the defects at asteroids fly-bys are relatively small (e.g., few hundreds m/s) and ASTRA solutions lie close to fully refined solutions, as discussed in Chapter 5.

Final lines 55-58 are for representation purposes. The presented script is useful to generate the path shown in Figure C.11. The spacecraft reaches Jupiter after 6.152 years, with $v_{\infty,dep} = 3.706$ km/s and $v_{\infty,arr} = 5.658$ km/s, while passing by 3 asteroids. The overall Δv needed to visit them is 0.2984 km/s.

Appendix D

Saturn Moons' Tours with Dynamic Programming

As further stress test for AUTOMATE, a long MGA sequence is searched within the Saturn system. Assuming a departing orbit around Saturn, one looks for a moon tour that ends at Enceladus with relatively low infinity velocity (e.g. 500 m/s). The added complexity of such mission scenario arises from the very high number of fly-by and DSMs options that are needed to achieve the declared goal. This is mainly because of the low gravity field of Saturn moons, thus providing limited orbit change when performing fly-bys.

SODP is used in this case to look for minimum Δv paths on Tisserand map [1]. This is implemented as described in Chapter 5. The nodes that are relevant for SODP application are triples of numbers made by (v_∞, α, k) as described in Chapter 4. Among all the paths that arrive at a common node, only the one with minimum accumulated Δv is kept for further expansion.

[1] Also MODP can be used, here SODP is implemented for the sake of simplicity.

D.1 Capture

As described in section C.1, the search is initiated by defining a starting node. For example, this can be an elliptical orbit within Saturn SOI that is reached after a Δv manoeuvre from the hyperbolic approach at the planet. Among Saturn moons, Titan presents a good option for initial orbit energy reduction, as it has the strongest gravity field the other moons considered here, thus allowing for more flexibility in the design process. The aim would then be to reach Titan at relatively low speed, e.g., $v_\infty = 2.8$ km/s to start the moon tour.

The strategy to reach such condition could be to:

- Arrive within Saturn SOI with a hyperbolic path (depending upon the spacecraft-Saturn relative velocity at the SOI).
- Perform a first manoeuvre (Δv_1) to move the spacecraft from the hyperbola to an elliptical capture orbit within the SOI.
- Perform a second manoeuvre (Δv_2) to arrive at Titan at $v_\infty = 2.8$ km/s.

The total Δv for the two manoeuvres would thus be function of:

- The spacecraft-Saturn relative velocity at the moment of encounter (here 5.5 km/s is considered, that is similar to the Cassini mission [171]).
- The size of the elliptical capture orbit around Saturn, namely eccentricity and periapsis (here a fixed eccentricity $e = 0.99$ is used, similar to the one used for Cassini, and the r_p is left free).

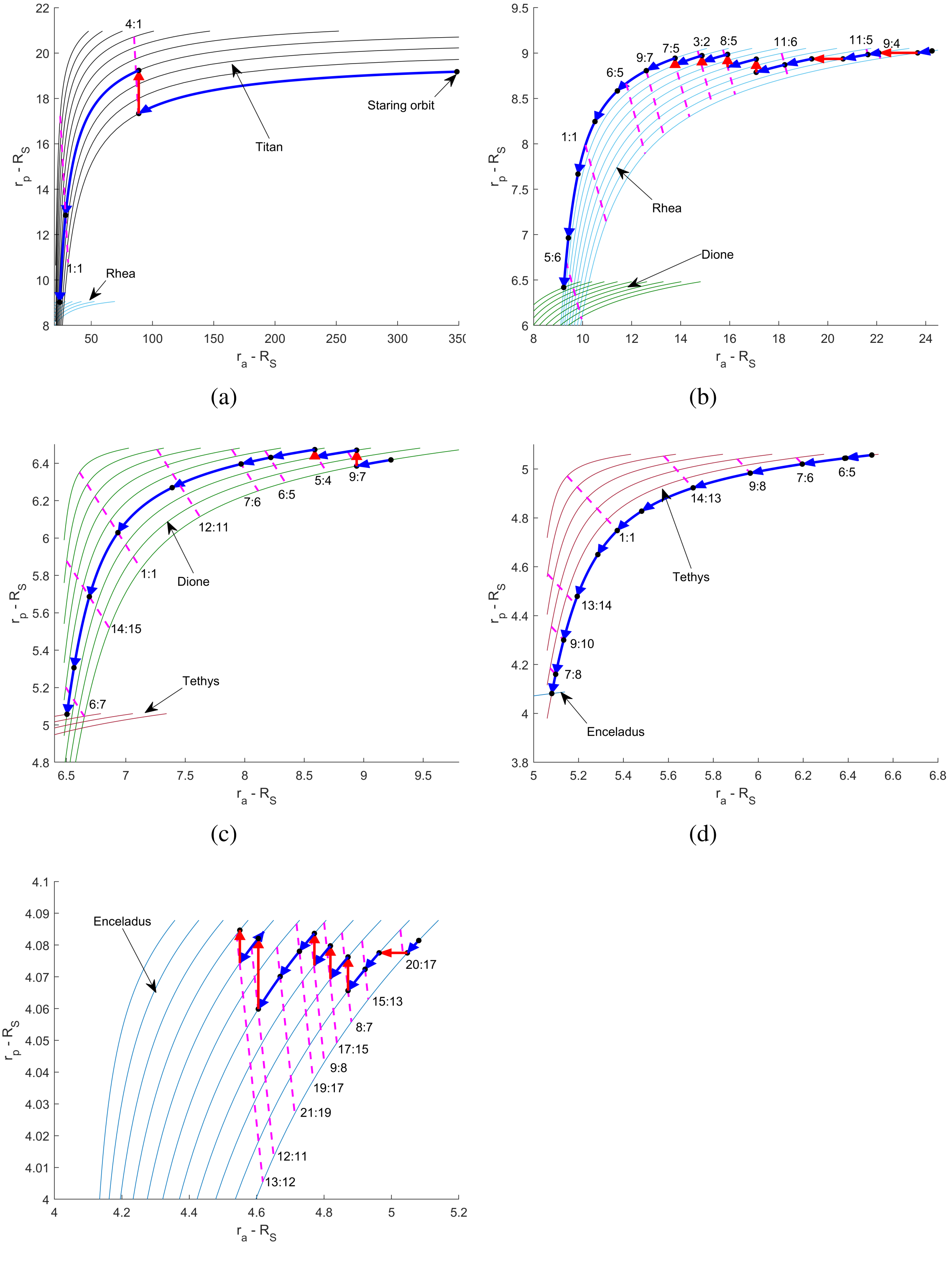
A good compromise is found to be $(r_a, r_p) = (348.37, 19.17)$ R_S (R_S being Saturn radius), allowing to reach Titan at $v_\infty = 2.8$ km/s with a total $\Delta v = \Delta v_1 + \Delta v_2 = 1.3$ km/s.

D.2 Moons' Tour

After the capture phase, the moons' tour can be initiated. Both fly-bys and VILTs are allowed, as described in Chapter 4. AUTOMATE scans the whole search space in terms of moon fly-bys and VILTs to construct sequences towards Enceladus. The optimization is divided in phases, one for each moon, on which the spacecraft performs fly-bys and VILTs with one single moon before moving to the next one. In this case, a single objective is optimized, that is the sum of the DSMs needed to reach Enceladus with infinity velocity of 500 m/s. The tour found by AUTOMATE is represented on Tisserand graphs in Figure D.1.

The first phase is the Titan (T) one, that allows to reach Rhea (R) with v_∞ = 1.8 km/s relative to Rhea on an overall TTTR sequence. From the starting orbit, the spacecraft performs a fly-by with Titan to reduce the apoapsis, reaching a quasi-resonant 4:1 orbit with the moon. A DSM of 0.0681 m/s is then performed to reduce the infinity velocity at Titan from 2800 m/s to 2200 m/s. Then, two more fly-bys are needed, the second one of which is in 1:1 resonance with Titan. The time needed to complete this phase is 79.65 days. The second phase is the one with Rhea. This is the longest one on the overall tour in terms of number of fly-bys and transfer time, with 14 encounters with Rhea and an overall 428.77 days of transfers. The overall Δv cost of the Rhea phase is 0.283 km/s. This allows to reduce the infinity velocity with respect to Rhea down to 1 km/s. After the last Rhea encounter, Dione is encountered with a relative velocity of 1 km/s. On the third phase, Dione is used to leverage the infinity velocity up to 0.7 km/s, using an overall Δv of 0.0442 km/s and 164.49 days of transfer. The spacecraft then reaches Tethys at a relative speed of 0.7 km/s. Interestingly, only resonant and quasi-resonant transfers are used at this moon, i.e., $\Delta v = 0$ km/s, with an overall 139.77 days of moon phase. Last phase is at Enceladus, on which only VILTs are used to leverage the v_∞ down to 0.350 km/s after 201.70 days and a cost of 0.105 km/s.

Although Tisserand graphs provide no explicit information on the transfer time between



(a) (b) (c) (d) (e)

Figure D.1: Tour phases at Titan (a), Rhea (b), Tethys (c), Dione (d) and Enceladus (e).

different moons' phases, as this depends upon the phasing between the objects, one could easily look for individual transfer strategies once a transfer option from Earth to Saturn is identified in terms of launch and arrival dates. In any case, one should not expect large variations in the overall transfer time within the moons' tour (few tens of days), due to the very low orbital periods of the objects.

Overall, the moon tour uses:

- 49 fly-bys with Saturnian moons
- 0.501 km/s of Δv
- Approximately 1018.51 days of transfer time

Appendix E

Asteroids Families, Regions and Special Targets

From Chapter 6, a pruned database of $\sim 102,000$ asteroids is used [9, 10] to assess the feasibility of a CASTAway-like mission. Asteroids' orbits that belong to such catalogue are represented in Figure E.1.

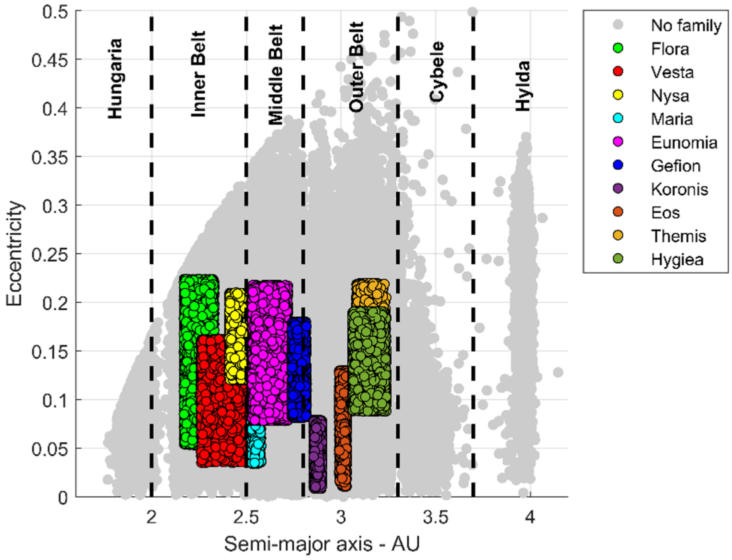


Figure E.1: Map of semi-major axis and eccentricity for the selected asteroids.

This shows semi-major axis and eccentricity of the objects and reports the main families and regions that are relevant for the mission analysis of CASTAway within the context of ESA's M-class call 2022.

The author also thanks Prof. Colin Snodgrass for providing a list of scientifically compelling asteroids. The list is reported in Table E.1.

Table E.1: List of scientific interesting asteroids for CASTAway-like missions.

Category 1: individual objects of special interest		
Number	**Name**	**Justification**
24	Themis	Possible water ice on surface from ground-based spectroscopy.
596	Scheila	Recent craters formed.
16	Psyche	Lump of metal. NASA is considering dedicated discovery mission.
65	Cybele	Possible water ice on surface.
Category 2: previous missions' targets		
Number	**Name**	**Justification**
1	Ceres	Surface evolution ∼ 20 years after Dawn [264]. Active body.
4	Vesta	Surface evolution ∼ 20 years after Dawn [264].
21	Lutetia	Surface evolution ∼ 25 years after Rosetta. [249].
Category 3: extreme sizes		
Number	**Name**	**Justification**
2	Pallas	Big size and interesting at 3 micron spectrum.
10	Hygiea	Big size and dark (7% albedo).
Category 4: extreme shapes		
Number	**Name**	**Justification**
216	Kleopatra	Highly elongated.
824	Anastasia	Elongated.
1248	Jugurtha	
208150	2000 FJ34	
211895	2004 JX32	
Category 5.a: family membership		
Number	**Name**	**Justification**
298	Baptistina	Possible source of Earth impactors.
396	Aeolia	Object from families < 100 Myr old. Largest asteroid of 306 family members.
606	Brangane	Object from families < 100 Myr old. Largest asteroid of 192 family members.
832	Karin	Very young family (∼ 6 Myr).
Category 5.b: possibly recently split		
Number	**Name**	
1270	Datura	
4765	Wasserburg	
5026	Martes	
6070	Rheinland	

Continued on next page

Table E.1 – *Continued from previous page*

6825	Irvine
7343	Ockeghem
8898	Linnaea
9068	1993 OD
9783	Tensho-kan
10123	Fideoja
10321	Rampo
10484	Hecht
11842	Kap'bos
13046	Aliev
13653	Priscus
15107	Toepperwein
15501	Pepawlowski
17198	Gorjup
17288	2000 NZ10
19289	1996 HY12
21436	Chaoyichi
21509	Lucascavin
21930	1999 VP61
22280	1985 CD2
23891	1998 SC49
23998	1999 RP29
25884	2000 SQ4
26416	1999 XM84
29358	1996 AY7
32957	1996 HX20
34162	2000 QV27
34380	2000 RV55
38184	1999 KF
38395	1999 RR193
39991	1998 HR37
40366	1999 NF27
40837	1999 TX95
42946	1999 TU95
48281	2002 EN153
48652	1995 VB
51161	2000 HY57
51609	2001 HZ32
52478	1995 TO
52773	1998 QU12
52852	1998 RB75
53754	2000 ED69
54041	2000 GQ113
56048	1998 XV39
56232	1999 JM31
57738	2001 UZ160
60151	1999 UZ6
60546	2000 EE85
60744	2000 GB93
63440	2001 MD30

Continued on next page

Table E.1 – *Continued from previous page*

63468	2001 OY21
64092	2001 SM289
65801	1996 AJ7
69142	2003 FL115
70511	1999 TL103
71484	2000 BE34
76111	2000 DK106
80218	1999 VO123
81337	2000 GP36
84203	2002 RD133
87887	2000 SS286
88259	2001 HJ7
89714	2001 YA114
92336	2000 GY81
92652	2000 QX36
99052	2001 ET15
101065	1998 RV11
101703	1999 CA150
106598	2000 WZ112
106700	2000 WX167
111335	2001 XL94
112249	2002 LM9
114243	2002 WZ5
118645	2000 JC14
128637	2004 RK22
133220	2003 QX79
138938	2001 BJ18
139537	2001 QE25
143155	2002 XS50
145314	2005 LE5
159435	1999 VJ178
165389	2000 WC188
180906	2005 KB6
183137	2002 RW219
184300	2005 ED114
189994	2004 GH33
195479	2002 GX130
203370	2001 WY35
220015	2002 PU155
226268	2003 AN55
226877	2004 TD93
228674	2002 JZ80
229401	2005 SU152
237517	2000 SP31
237738	2001 XN74
268305	2005 QV114
287982	2003 UQ164
318689	2005 QC62
337108	1999 RV84
338178	2002 RJ126

Continued on next page

Table E.1 – *Continued from previous page*

344172	2001 AV49
Category 6.a: triple	
Number	**Name**
45	Eugenia
87	Sylvia
93	Minerva
130	Elektra
216	Kleopatra
Category 6.b: binary	
22	Kalliope
41	Daphne
90	Antiope
107	Camilla
121	Hermione
243	Ida and Dactyl
283	Emma
317	Roxane
379	Huenna
702	Alauda
762	Pulcova
809	Lundia
854	Frostia
939	Isberga
1052	Belgica
1089	Tama
1453	Fennia
1509	Esclangona
1717	Arlon
1830	Pogson
2006	Polonskaya
2047	Smetana
2121	Sevastopol
2131	Mayall
2343	Siding Spring
2478	Tokai
2486	Metsahovi
2623	Zech
2691	Sersic
2754	Efimov
2815	Soma
3034	Climenhaga
3073	Kursk
3169	Ostro
3309	Brorfelde
3433	Fehrenbach
3673	Levy
3703	Volkonskaya
3782	Celle
3841	Dicicco
3868	Mendoza

Continued on next page

Table E.1 – *Continued from previous page*

3905	Doppler
3951	Zichichi
3982	Kastel
4029	Bridges
4272	Entsuji
4383	Suruga
4440	Tchantches
4492	Debussy
4514	Vilen
4541	Mizuno
4607	Seilandfarm
4666	Dietz
4674	Pauling
4765	Wasserburg
4786	Tatianina
4868	Knushevia
4951	Iwamoto
5425	Vojtech
5426	Sharp
5474	Gingasen
5477	Holmes
5481	Kiuchi
5899	Jedicke
5905	Johnson
6084	Bascom
6244	Okamoto
6265	1985 TW3
6369	1983 UC
6615	Plutarchos
6708	Bobbievaile
7187	Isobe
7225	Huntress
7958	Leakey
8026	Johnmckay
8116	Jeanperrin
8306	Shoko
8474	Rettig
9069	Hovland
9260	Edwardolson
9617	Grahamchapman
9783	Tensho-kan
10123	Fideoja
10208	Germanicus
11217	1999 JC4
11264	Claudiomaccone
13123	Tyson
15268	Wendelinefroger
15430	1998 UR31
15822	1994 TV15
16525	Shumarinaiko

Continued on next page

Table E.1 – *Continued from previous page*

17246	2000 GL74
17260	2000 JQ58
18890	2000 EV26
20325	1998 HO27
21436	Chaoyichi
22899	1999 TO14
26416	1999 XM84
27568	2000 PT6
31450	1999 CU9
32008	2000 HM53
43008	1999 UD31
44620	1999 RS43
46829	1998 OS14
52316	1992 BD
69406	1995 SX48
76818	2000 RG79
79472	1998 AX4

References

[1] R. S. Johnston and W. Hull, “Apollo missions,” *Biomedical Results of Apollo, eds RS Johnston, LF Dietlein and C. Berry (Washington, DC: National Aeronautics and Space Administration)*, pp. 9–40, 1975.

[2] R. Reinhard, “The Giotto encounter with comet Halley,” *Nature*, vol. 321, no. 6067, pp. 313–318, 1986. `https://doi.org/10.1038/321313a0`.

[3] J. A. Dunne and E. Burgess, “The voyage of Mariner 10,” *National Aeronautics and Space Administration SP*, vol. 424, 1978.

[4] R. Schmidt, “Mars Express—ESA’s first mission to planet Mars,” *Acta Astronautica*, vol. 52, no. 2-6, pp. 197–202, 2003. `https://doi.org/10.1016/S0094-5765(02)00157-1`.

[5] C. Hall, “Pioneer 10/11 spacecraft and missions to Jupiter,” in *Amsterdam International Astronautical Federation Congress*, 1974.

[6] O. Grasset, M. K. Dougherty, A. Coustenis, E. J. Bunce, C. Erd, D. Titov, M. Blanc, A. Coates, P. Drossart, and L. N. Fletcher, “JUpiter ICy moons Explorer (JUICE): An ESA mission to orbit Ganymede and to characterise the Jupiter system,” *Planetary and Space Science*, vol. 78, pp. 1–21, 2013. `https://doi.org/10.1016/j.pss.2012.12.002`.

[7] W.-J. Li, D.-Y. Cheng, X.-G. Liu, Y.-B. Wang, W.-H. Shi, Z.-X. Tang, F. Gao, F.-M. Zeng, H.-Y. Chai, W.-B. Luo, *et al.*, “On-orbit service (OOS) of spacecraft: A

review of engineering developments," *Progress in Aerospace Sciences*, vol. 108, pp. 32–120, 2019. https://doi.org/10.1016/j.paerosci.2019.01.004.

[8] C. P. Mark and S. Kamath, "Review of active space debris removal methods," *Space Policy*, vol. 47, pp. 194–206, 2019. https://doi.org/10.1016/j.spacepol.2018.12.005.

[9] J. P. Sánchez, A. Gibbings, C. Snodgrass, S. Green, and N. Bowles, "Asteroid belt multiple fly-by options for M-class missions," in *Proceedings of the International Astronautical Congress*, 2016.

[10] N. E. Bowles, C. Snodgrass, A. Gibbings, J. P. Sánchez, J. A. Arnold, P. Eccleston, T. Andert, A. Probst, G. Naletto, and A. C. Vandaele, "CASTAway: An asteroid main belt tour and survey," *Advances in space research*, vol. 62, no. 8, pp. 1998–2025, 2018. https://doi.org/10.1016/j.asr.2017.10.021.

[11] J. A. Englander, K. Berry, B. Sutter, D. Stanbridge, D. H. Ellison, K. Williams, J. McAdams, J. M. Knittel, C. Welch, and H. Levison, "Trajectory design of the Lucy mission to explore the diversity of the Jupiter Trojans," in *Proceedings of the International Astronautical Congress*, 2019.

[12] J. R. Wertz, D. F. Everett, and J. J. Puschell, *Space mission engineering: the new SMAD*. Microcosm Press, 2011.

[13] J. Schoenmaekers, R. Jehn, M. Landgraf, and M. Khan, "Mission Analysis - Towards an European Harmonisation," 2008. Available online at: https://www.esa.int/esapub/bulletin/bulletin134/bul134b_schoenmaekers.pdf, Last accessed: October 2022.

[14] A. Shirazi, J. Ceberio, and J. A. Lozano, "Spacecraft trajectory optimization: A review of models, objectives, approaches and solutions," *Progress in Aerospace Sciences*, vol. 102, pp. 76–98, 2018. https://doi.org/10.1016/j.paerosci.2018.07.007.

[15] H. Curtis, *Orbital mechanics for engineering students*. Butterworth-Heinemann, 2013.

[16] D. A. Vallado, *Fundamentals of astrodynamics and applications*, vol. 12. Springer Science & Business Media, 2001.

[17] J. Sims and S. Flanagan, "Preliminary design of low-thrust interplanetary missions," in *Proceedings of AAS/AIAA Astrodynamics Specialists Conference*, 1999.

[18] P. Gurfil, *Modern astrodynamics*. Elsevier, 2006.

[19] V. Kumar, "Algorithms for constraint-satisfaction problems: A survey," *AI magazine*, vol. 13, pp. 32–44, 1992. https://doi.org/10.1609/aimag.v13i1.976.

[20] S. C. Brailsford, C. N. Potts, and B. M. Smith, "Constraint satisfaction problems: Algorithms and applications," *European journal of operational research*, vol. 119, no. 3, pp. 557–581, 1999. https://doi.org/10.1016/S0377-2217(98)00364-6.

[21] D. E. Kirk, *Optimal control theory: an introduction*. Courier Corporation, 2004. https://doi.org/10.1002/aic.690170452.

[22] I. M. Ross, *A primer on Pontryagin's principle in optimal control*. Collegiate publishers, 2015.

[23] J. T. Betts, "Survey of numerical methods for trajectory optimization," *Journal of guidance, control, and dynamics*, vol. 21, no. 2, pp. 193–207, 1998. https://doi.org/10.2514/2.4231.

[24] B. A. Conway, "A survey of methods available for the numerical optimization of continuous dynamic systems," *Journal of Optimization Theory and Applications*, vol. 152, no. 2, pp. 271–306, 2012. https://doi.org/10.1007/s10957-011-9918-z.

[25] S. Kemble, *Interplanetary mission analysis and design*. Springer Science & Busi-

ness Media, 2006. https://doi.org/10.1007/3-540-37645-3.

[26] M. Diehl, H. G. Bock, H. Diedam, and P.-B. Wieber, "Fast direct multiple shooting algorithms for optimal robot control," in *Fast motions in biomechanics and robotics*, pp. 65–93, Springer, 2006. https://doi.org/10.1007/978-3-540-36119-0_4.

[27] J. Z. Ben-Asher, *Optimal control theory with aerospace applications*. American Institute of Aeronautics and Astronautics, 2010. https://doi.org/10.2514/4.867347.

[28] C. Blum and A. Roli, "Metaheuristics in combinatorial optimization: Overview and conceptual comparison," *ACM computing surveys (CSUR)*, vol. 35, no. 3, pp. 268–308, 2003. https://doi.org/10.1145/937503.937505.

[29] S. Voß, S. Martello, I. H. Osman, and C. Roucairol, *Meta-heuristics: Advances and Trends in Local Search Paradigms for Optimization*. Springer Science & Business Media, 2012. https://doi.org/10.1007/978-1-4615-5775-3.

[30] B. Peherstorfer, K. Willcox, and M. Gunzburger, "Survey of multifidelity methods in uncertainty propagation, inference, and optimization," *Siam Review*, vol. 60, no. 3, pp. 550–591, 2018. https://doi.org/10.1137/16M1082469.

[31] M. Ceriotti, *Global optimisation of multiple gravity assist trajectories*. PhD thesis, University of Glasgow, 2010.

[32] A. Bellome, J. P. Sánchez, J. I. Rico Álvarez, H. Afsa, S. Kemble, and L. Felicetti, "An Automatic Process for Sample Return Missions Based on Dynamic Programming Optimization," in *AIAA SCITECH 2022 Forum*, p. 1477, 2022. https://doi.org/10.2514/6.2022-1477.

[33] C. A. Kluever, *Space flight dynamics*. John Wiley & Sons, 2018.

[34] J. T. Betts, *Practical methods for optimal control and estimation using nonlinear*

programming. SIAM, 2010. https://doi.org/10.1137/1.9780898718577.

[35] B. A. Conway, *Spacecraft trajectory optimization*. Cambridge University Press, 2010. https://doi.org/10.1017/CBO9780511778025.

[36] A. V. Rao, "A survey of numerical methods for optimal control," *Advances in the Astronautical Sciences*, vol. 135, no. 1, pp. 497–528, 2009.

[37] R. H. Battin, "Lambert's problem revisited," *AIAA Journal*, vol. 15, no. 5, pp. 707–713, 1977. https://doi.org/10.2514/3.60680.

[38] R. H. Battin, *An Introduction to the Mathematics and Methods of Astrodynamics, revised edition*. American Institute of Aeronautics and Astronautics, 1999. https://doi.org/10.2514/4.861543.

[39] R. H. Gooding, "A procedure for the solution of Lambert's orbital boundary-value problem," *Celestial Mechanics and Dynamical Astronomy*, vol. 48, no. 2, pp. 145–165, 1990. https://doi.org/10.1007/BF00049511.

[40] D. Izzo, "Revisiting Lambert's problem," *Celestial Mechanics and Dynamical Astronomy*, vol. 121, no. 1, pp. 1–15, 2015. https://doi.org/10.1007/s10569-014-9587-y.

[41] Á. B. Jiménez, J. L. Lázaro, and J. R. Dorronsoro, "Finding optimal model parameters by discrete grid search," in *Innovations in Hybrid Intelligent Systems*, pp. 120–127, Springer, 2007. https://doi.org/10.1007/978-3-540-74972-1_17.

[42] M. Grötschel, M. W. Padberg, E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, and D. B. Schmoys, *The traveling salesman problem*. New York: Wiley, 1985.

[43] A. H. Land and A. G. Doig, "An automatic method for solving discrete programming problems," *Econometrica*, vol. 28, pp. 497–520, 1960. https://doi.org/10.1007/978-3-540-68279-0_5.

[44] S. C. Shapiro, *Encyclopedia of artificial intelligence second edition*. John Wiley

and Sons, 1992.

[45] D. Izzo, I. Getzner, D. Hennes, and L. F. Simões, "Evolving solutions to TSP variants for active space debris removal," in *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, pp. 1207–1214, 2015. https://doi.org/10.1145/2739480.2754727.

[46] M. Schlueter, *Nonlinear mixed integer based optimization technique for space applications*. PhD thesis, University of Birmingham, 2012.

[47] M. Schlueter, S. O. Erb, M. Gerdts, S. Kemble, and J.-J. Rückmann, "MIDACO on MINLP space applications," *Advances in Space Research*, vol. 51, no. 7, pp. 1116–1131, 2013. https://doi.org/10.1016/j.asr.2012.11.006.

[48] I. M. Ross and C. N. D'Souza, "Hybrid optimal control framework for mission planning," *Journal of Guidance, Control, and Dynamics*, vol. 28, no. 4, pp. 686–697, 2005. https://doi.org/10.2514/1.8285.

[49] D. Izzo, "1st ACT global trajectory optimisation competition: Problem description and summary of the results," *Acta Astronautica*, vol. 61, no. 9, pp. 731–734, 2007. https://doi.org/10.1016/j.actaastro.2007.03.003.

[50] A. Törn, M. M. Ali, and S. Viitanen, "Stochastic global optimization: Problem classes and solution techniques," *Journal of Global Optimization*, vol. 14, no. 4, pp. 437–447, 1999. https://doi.org/10.1023/A:1008395408187.

[51] L. Casalino, G. Colasurdo, and M. R. Sentinella, "Indirect optimization method for low-thrust interplanetary trajectories," in *30th International Electric Propulsion Conference, Florence*, Citeseer, 2007.

[52] I. Grigoriev and M. Zapletin, "Choosing promising sequences of asteroids.," *Automation & Remote Control*, vol. 74, no. 8, 2013. https://doi.org/10.1134/S0005117913080055.

[53] A. E. Petropoulos, E. P. Bonfiglio, D. J. Grebow, T. Lam, J. S. Parker, J. Arrieta, D. F. Landau, R. L. Anderson, E. D. Gustafson, and G. J. Whiffen, "GTOC5: results from the Jet Propulsion Laboratory," *Acta Futura*, vol. 8, pp. 21–27, 2014. `https://doi.org/10.2420/AF08.2014.21`.

[54] G. Colasurdo, A. Zavoli, A. Longo, L. Casalino, and F. Simeoni, "Tour of Jupiter Galilean moons: Winning solution of GTOC6," *Acta Astronautica*, vol. 102, pp. 190–199, 2014. `https://doi.org/10.1016/j.actaastro.2014.06.003`.

[55] J. E. Petropoulos, G. Lantoine, D. Landau, D. Grebow, T. Lam, N. Arora, D. Jones, J. Senent, M. Jesick, J. Arrieta, A. Zimmer, E. Gustafson, J. Bellerose, T. Pavlak, M. Vaquero, J. Stuart, G. Whiffen, P. Finlayson, J. Sims, "Team 19 Response to the 7th Global Trajectory Optimisation Competition hosted by Politecnico di Torino and Universit'a di Roma, La Sapienza," 2014. `https://sophia.estec.esa.int/gtoc_portal/wp-content/uploads/2014/09/team19.pdf`, Last accessed: September 2022.

[56] D. Izzo, D. Hennes, M. Märtens, I. Getzner, K. Nowak, A. Heffernan, S. Campagnola, C. H. Yam, N. Ozaki, and Y. Sugimoto, "GTOC8: Results and Methods of ESA Advanced Concepts Team and JAXA-ISAS," in *Proceedings of AAS/AIAA Astrodynamics Specialist Conference*, 2016.

[57] A. E. Petropoulos, D. Grebow, D. Jones, G. Lantoine, A. Nicholas, J. Roa, J. Senent, J. Stuart, N. Arora, T. Pavlak, *et al.*, "GTOC9: Results from the Jet Propulsion Laboratory team," *Acta Futura*, vol. 11, pp. 25–35, 2018. `https://doi.org/10.5281/zenodo.1139152`.

[58] Y.-Z. Luo, H.-X. Shen, A.-Y. Huang, T.-J. Zhang, Y.-H. Zhu, Z. Li, P. Shu, Z.-J. Sun, J.-H. Li, and Z.-Y. Li, "GTOC X: Results and methods of National University of Defense Technology and Xi'an Satellite Control Center," in *Proceedings of AAS/AIAA Astrodynamics Specialist Conference*, 2019.

[59] Z. Zhang, N. Zhang, X. Guo, D. Wu, X. Xie, J. Li, J. Yang, S. Chen, F. Jiang, H. Baoyin, *et al.*, "GTOC 11: Results from Tsinghua University and Shanghai Institute of Satellite Engineering," *Acta Astronautica*, 2022. `https://doi.org/10.1016/j.actaastro.2022.06.028`.

[60] K. Alemany and R. D. Braun, "Survey of global optimization methods for low-thrust, multiple asteroid tour missions," in *17th AAS/AIAA Space Flight Mechanics Meeting*, 2007.

[61] M. Di Carlo, M. Vasile, and J. Dunlop, "Low-thrust tour of the main belt asteroids," *Advances in Space Research*, vol. 62, no. 8, pp. 2026–2045, 2018. `https://doi.org/10.1016/j.asr.2017.12.033`.

[62] R. Armellin, L. Beauregard, A. Bellome, N. Bernardini, A. Fossa, X. Fu, H. Holt, C. Parigini, L. Pirovano, and M. Wijayatunga, "Team theAntipodes: Solution methodology for GTOC11," *Acta Astronautica*, vol. 201, pp. 142–151, 2022. `https://doi.org/10.1016/j.actaastro.2022.08.034`.

[63] F. Jiang, Y. Chen, Y. Liu, H. Baoyin, and J. Li, "GTOC5: results from the Tsinghua University," *Acta Futura*, vol. 8, pp. 37–44, 2014. `http://dx.doi.org/10.2420/AF08.2014.37`.

[64] A. H. G. E. Gad, *Space trajectories optimization using variable-chromosome-length genetic algorithms*. PhD thesis, Michigan Technological University, 2011.

[65] D. Izzo, L. F. Simoes, C. H. Yam, F. Biscani, D. Di Lorenzo, B. Addis, and A. Cassioli, "GTOC5: results from the European Space Agency and University of Florence," *Acta Futura*, vol. 8, pp. 45–55, 2014. `http://dx.doi.org/10.2420/AF08.2014.45`.

[66] D. J. Kessler and B. G. Cour-Palais, "Collision frequency of artificial satellites: The creation of a debris belt," *Journal of Geophysical Research: Space Physics*, vol. 83, no. A6, pp. 2637–2646, 1978. `https://doi.org/10.1029/JA083iA06p02637`.

[67] D. J. Kessler, N. L. Johnson, J. C. Liou, and M. Matney, "The Kessler syndrome: implications to future space operations," *Advances in the Astronautical Sciences*, vol. 137, no. 8, p. 2010, 2010.

[68] D. Izzo and M. Märtens, "The Kessler run: on the design of the GTOC9 challenge," *Acta Futura*, vol. 11, pp. 11–24, 2018. `https://doi.org/10.5281/zenodo.1139022`.

[69] H.-X. Shen, T.-J. Zhang, A. Huang, and Z. Li, "GTOC9: Results from the Xi'an Satellite Control Center (team XSCC)," *Acta Futura*, vol. 11, pp. 49–55, 2018. `https://doi.org/10.5281/zenodo.1139240`.

[70] Y.-Z. Luo, Y. Zhu, H. Zhu, Z. Yang, Z.-J. Sun, and J. Zhang, "GTOC9: Results from the National University of Defense Technology (team NUDT)," *Acta Futura*, vol. 11, pp. 37–47, 2018. `https://doi.org/10.5281/zenodo.1139226`.

[71] J. Yunta Gurumeta, *Multi-target Rendezvous Combinatorial Optimisation Applied to GTOC9.* Master thesis, Cranfield University, 2020.

[72] H. Li, S. Chen, D. Izzo, and H. Baoyin, "Deep networks as approximators of optimal low-thrust and multi-impulse cost in multitarget missions," *Acta Astronautica*, vol. 166, pp. 469–481, 2020. `https://doi.org/10.1016/j.actaastro.2019.09.023`.

[73] H.-X. Shen and L. Casalino, "Simple ΔV Approximation for Optimization of Debris-to-Debris Transfers," *Journal of Spacecraft and Rockets*, vol. 58, no. 2, pp. 575–580, 2021. `https://doi.org/10.2514/1.A34831`.

[74] A. Bellome, "ASTRA_TOOLBOXES," Dec 2022. `https://doi.org/10.17862/cranfield.rd.21747758`.

[75] V. Sterken, "The DOLPHIN mission and unique opportunities in 2030 to probe the dust-heliosphere interactions," *44th COSPAR Scientific Assembly. Held 16-24 July*, vol. 44, p. 1016, 2022.

[76] T. Lehtinen, M. Granvik, A. Bellome, and J. P. Sánchez, "Icarus: In-situ monitoring of the surface degradation on a near-sun asteroid," *Acta Astronautica*, vol. 186, pp. 98–108, 2021. https://doi.org/10.1016/j.actaastro.2021.05.028.

[77] ESA, "Call for a Medium-Size and a Fast Mission Opportunity in ESA's Science Programme - Technical Annex (ESA-SCI-F-MGT-MAN-001)," 2021. Available online at: https://www.cosmos.esa.int/documents/7423467/7423486/M7andF2_call_technical_annex_211211_final.pdf/460941d9-e57e-e201-6ac3-53753016d21d?t=1639389586597, Last accessed: February 2022.

[78] R. Armellin, D. Gondelach, and J. F. San Juan, "Multiple revolution perturbed Lambert problem solvers," *Journal of Guidance, Control, and Dynamics*, vol. 41, no. 9, pp. 2019–2032, 2018. https://doi.org/10.2514/1.G003531.

[79] D. Folta and D. Quinn, "Lunar frozen orbits," *Collection of Technical Papers - AIAA/AAS Astrodynamics Specialist Conference, 2006*, vol. 3, no. August, pp. 1915–1932, 2006.

[80] D. Karydis, G. Voyatzis, and K. Tsiganis, "A continuation approach for computing periodic orbits around irregular-shaped asteroids. An application to 433 Eros," *Advances in Space Research*, vol. 68, no. 11, pp. 4418–4433, 2021. https://doi.org/10.1016/j.asr.2021.08.036.

[81] Y. Ulybyshev, "Long-term station keeping of space station in lunar halo orbits," *Journal of Guidance, Control, and Dynamics*, vol. 38, no. 6, pp. 1063–1070, 2015. https://doi.org/10.2514/1.G000242.

[82] W. S. Koon, M. W. Lo, J. E. Marsden, and S. D. Ross, "Dynamical systems, the three-body problem and space mission design," in *Equadiff 99: (In 2 Volumes)*, pp. 1167–1181, World Scientific, 2000. https://doi.org/10.1142/9789812792617_0222.

[83] E. J. Doedel, V. A. Romanov, R. C. Paffenroth, H. B. Keller, D. J. Dichmann, J. Galán-Vioque, and A. Vanderbauwhede, "Elemental periodic orbits associated with the libration points in the circular restricted 3-body problem," *International Journal of Bifurcation and Chaos*, vol. 17, no. 08, pp. 2625–2677, 2007. `https://doi.org/10.1142/S0218127407018671`.

[84] A. I. McInnes, "Strategies for solar sail mission design in the circular restricted three-body problem," *MSE Thesis, School of Aeronautics and Astronautics, Purdue University*, 2000.

[85] S. Campagnola, P. Skerritt, and R. P. Russell, "Flybys in the planar, circular, restricted, three-body problem," *Celestial Mechanics and Dynamical Astronomy*, vol. 113, no. 3, pp. 343–368, 2012. `https://doi.org/10.1007/s10569-012-9427-x`.

[86] R. Thurman and P. A. Worfolk, "The geometry of halo orbits in the circular restricted three-body problem," *University of Minnesota: Geometry Center Research Report GCG95*, 1996.

[87] G. R. Hintz, "Survey of orbit element sets," *Journal of guidance, control, and dynamics*, vol. 31, no. 3, pp. 785–790, 2008. `https://doi.org/10.2514/1.32237`.

[88] O. Abdelkhalik and D. Mortari, "N-impulse orbit transfer using genetic algorithms," *Journal of Spacecraft and Rockets*, vol. 44, no. 2, pp. 456–460, 2007. `https://doi.org/10.2514/1.24701`.

[89] O. Abdelkhalik and A. Gad, "Optimization of space orbits design for Earth orbiting missions," *Acta Astronautica*, vol. 68, no. 7-8, pp. 1307–1317, 2011. `https://doi.org/10.1016/j.actaastro.2010.09.029`.

[90] M. Vasile and F. Zuiani, "Multi-agent collaborative search: an agent-based memetic multi-objective optimization algorithm applied to space trajectory de-

sign," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 225, no. 11, pp. 1211–1227, 2011. https://doi.org/10.1177/0954410011410274.

[91] A. Shirazi, "Multi-objective optimization of orbit transfer trajectory using imperialist competitive algorithm," in *2017 IEEE Aerospace Conference*, pp. 1–14, IEEE, 2017. https://doi.org/10.1109/AERO.2017.7943921.

[92] A. E. Bryson and Y.-C. Ho, *Applied optimal control: optimization, estimation, and control.* Routledge, 2018. https://doi.org/10.1201/9781315137667.

[93] Y. Wang, M. Zhu, Y. Wei, and Y. Zhang, "Solar sail spacecraft trajectory optimization based on improved imperialist competitive algorithm," in *Proceedings of the 10th World Congress on Intelligent Control and Automation*, pp. 191–195, IEEE, 2012. https://doi.org/10.1109/WCICA.2012.6357865.

[94] H. Li, H. Baoyin, and F. Topputo, "Neural networks in time-optimal low-thrust interplanetary transfers," *IEEE Access*, vol. 7, pp. 156413–156419, 2019. https://doi.org/10.1109/ACCESS.2019.2946657.

[95] M. Vasile and M. Locatelli, "A hybrid multiagent approach for global trajectory optimization," *Journal of Global Optimization*, vol. 44, no. 4, pp. 461–479, 2009. https://doi.org/10.1007/s10898-008-9329-3.

[96] Y.-Z. Luo, G.-J. Tang, Y.-J. Lei, and H.-Y. Li, "Optimization of multiple-impulse, multiple-revolution, rendezvous-phasing maneuvers," *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 4, pp. 946–952, 2007. https://doi.org/10.2514/1.25620.

[97] Y.-Z. Luo, J. Zhang, H.-y. Li, and G.-J. Tang, "Interactive optimization approach for optimal impulsive rendezvous using primer vector and evolutionary algorithms," *Acta Astronautica*, vol. 67, no. 3-4, pp. 396–405, 2010. https://doi.org/10.1016/j.actaastro.2010.02.014.

[98] M. Ceriotti and M. Vasile, "MGA trajectory planning with an ACO-inspired algorithm," *Acta Astronautica*, vol. 67, no. 9-10, pp. 1202–1217, 2010. https://doi.org/10.1016/j.actaastro.2010.07.001.

[99] H. Shang, H. Cui, P. Cui, and E. Luan, "Optimal-fuel, low-thrust Earth-Ivar transfer trajectories with Venus gravity assist," in *2006 1st International Symposium on Systems and Control in Aerospace and Astronautics*, pp. 5–pp, IEEE, 2006. https://doi.org/10.1109/ISSCAA.2006.1627643.

[100] H. Shang, P. Cui, E. Luan, and D. Qiao, "Study of design and optimization of low-thrust transfer trajectory with planetary aerogravity assist," in *The Proceedings of the Multiconference on" Computational Engineering in Systems Applications"*, vol. 1, pp. 690–695, IEEE, 2006. https://doi.org/10.1109/CESA.2006.4281741.

[101] L. W. Neustadt, "A general theory of minimum-fuel space trajectories," *Journal of the Society for Industrial and Applied Mathematics, Series A: Control*, vol. 3, no. 2, pp. 317–356, 1965. https://doi.org/10.1137/0303023.

[102] Q. Chen, D. Qiao, and C. Wen, "Minimum-Fuel Low-Thrust Trajectory Optimization via Reachability Analysis and Convex Programming," *Journal of Guidance, Control, and Dynamics*, vol. 44, no. 5, pp. 1036–1043, 2021. https://doi.org/10.2514/1.G004766.

[103] Y. Wang and F. Topputo, "Indirect Optimization of Power-Limited Asteroid Rendezvous Trajectories," *Journal of Guidance, Control, and Dynamics*, vol. 45, no. 5, pp. 962–971, 2022. https://doi.org/10.2514/1.G006179.

[104] S. Li, R. Mehra, R. Smith, and R. Beard, "Multi-spacecraft trajectory optimization and control using genetic algorithm techniques," in *2000 IEEE Aerospace Conference. Proceedings (Cat. No. 00TH8484)*, vol. 7, pp. 99–108, IEEE, 2000. https://doi.org/10.1109/AERO.2000.879279.

[105] M. Ceriotti and C. R. McInnes, “Hybrid solar sail and solar electric propulsion for novel earth observation missions,” *Acta Astronautica*, vol. 69, no. 9-10, pp. 809–821, 2011. `https://doi.org/10.1016/j.actaastro.2011.06.007`.

[106] F. L. Lewis, D. Vrabie, and V. L. Syrmos, *Optimal control*. John Wiley & Sons, 2012. `https://doi.org/10.1002/9781118122631`.

[107] R. T. Marler and J. S. Arora, “Survey of multi-objective optimization methods for engineering,” *Structural and multidisciplinary optimization*, vol. 26, no. 6, pp. 369–395, 2004. `https://doi.org/10.1007/s00158-003-0368-6`.

[108] C. Hillermeier, “Generalized homotopy approach to multiobjective optimization,” *Journal of Optimization Theory and Applications*, vol. 110, no. 3, pp. 557–583, 2001. `https://doi.org/10.1023/A:1017536311488`.

[109] A. M. Montesano, A. Zanni, and L. Bruni, *Vilfredo Pareto. Manuale di Economia Politica. Edizione critica*. Università Bocconi Editore, 2006.

[110] W. Tu and R. Mayne, “Studies of multi-start clustering for global optimization,” *International journal for numerical methods in engineering*, vol. 53, no. 9, pp. 2239–2252, 2002. `https://doi.org/10.1002/nme.400`.

[111] C. Snodgrass, G. H. Jones, H. Böhnhardt, A. Gibbings, M. Homeister, N. Andre, P. Beck, M. S. Bentley, I. Bertini, and N. Bowles, “The Castalia mission to main belt comet 133P/Elst-Pizarro,” *Advances in Space Research*, vol. 62, no. 8, pp. 1947–1976, 2018. `https://doi.org/10.1016/j.asr.2017.09.011`.

[112] M. Di Carlo, J. M. Romero Martin, and M. Vasile, “CAMELOT: computational-analytical multi-fidElity low-thrust optimisation toolbox,” *CEAS Space Journal*, vol. 10, no. 1, pp. 25–36, 2018. `https://doi.org/10.1007/s12567-017-0172-6`.

[113] M. Di Carlo and M. Vasile, “Analytical solutions for low-thrust orbit transfers,” *Celestial Mechanics and Dynamical Astronomy*, vol. 133, no. 7, pp. 1–38, 2021.

https://doi.org/10.1007/s10569-021-10033-9.

[114] A. A. Quarta and G. Mengali, "Semi-analytical method for the analysis of solar sail heliocentric orbit raising," *Journal of Guidance, Control, and Dynamics*, vol. 35, no. 1, pp. 330–335, 2012. https://doi.org/10.2514/1.55101.

[115] O. Von Stryk and R. Bulirsch, "Direct and indirect methods for trajectory optimization," *Annals of operations research*, vol. 37, no. 1, pp. 357–373, 1992. https://doi.org/10.1007/BF02071065.

[116] R. Bellman, "The theory of dynamic programming," *Bulletin of the American Mathematical Society*, vol. 60, no. 6, pp. 503–515, 1954. https://doi.org/10.1090/S0002-9904-1954-09848-8.

[117] L. S. Pontryagin, *Mathematical theory of optimal processes.* CRC press, 1987. https://doi.org/10.1201/9780203749319.

[118] R. Bellman and R. E. Kalaba, *Dynamic programming and modern control theory*, vol. 81. Citeseer, 1965.

[119] M. R. Caputo, "Dynamic Programming and the Hamilton-Jacobi-Bellman Equation," *Foundations of Dynamic Economic Analysis: Optimal Control Theory and Applications; Cambridge University Press: Cambridge, UK*, pp. 511–536, 2005. https://doi.org/10.1017/CBO9780511806827.

[120] D. Bertsekas, *Dynamic programming and optimal control*, vol. 1. Athena scientific, 2012. https://dl.acm.org/doi/10.5555/526680.

[121] S. Yakowitz and B. Rutherford, "Computational aspects of discrete-time optimal control," *Applied Mathematics and Computation*, vol. 15, no. 1, pp. 29–45, 1984. https://doi.org/10.1016/0096-3003(84)90051-1.

[122] C. Colombo, M. Vasile, and G. Radice, "Optimal low-thrust trajectories to asteroids through an algorithm based on differential dynamic programming," *Ce-*

lestial mechanics and dynamical astronomy, vol. 105, no. 1, pp. 75–112, 2009. https://doi.org/10.1007/s10569-009-9224-3.

[123] G. Lantoine and R. P. Russell, "A hybrid differential dynamic programming algorithm for constrained optimal control problems. Part 1: Theory," *Journal of Optimization Theory and Applications*, vol. 154, no. 2, pp. 382–417, 2012. https://doi.org/10.1007/s10957-012-0039-0.

[124] G. Lantoine and R. P. Russell, "A hybrid differential dynamic programming algorithm for constrained optimal control problems. Part 2: Application," *Journal of Optimization Theory and Applications*, vol. 154, no. 2, pp. 418–442, 2012. https://doi.org/10.1007/s10957-012-0038-1.

[125] D. H. Jacobson, "New second-order and first-order algorithms for determining optimal control: A differential dynamic programming approach," *Journal of Optimization Theory and Applications*, vol. 2, no. 6, pp. 411–440, 1968. https://doi.org/10.1007/BF00925746.

[126] D. H. Jacobson, "Differential dynamic programming methods for solving bang-bang control problems," *IEEE Transactions on Automatic Control*, vol. 13, no. 6, pp. 661–675, 1968. https://doi.org/10.1109/TAC.1968.1099026.

[127] D. A. Benson, G. T. Huntington, T. P. Thorvaldsen, and A. V. Rao, "Direct trajectory optimization and costate estimation via an orthogonal collocation method," *Journal of Guidance, Control, and Dynamics*, vol. 29, no. 6, pp. 1435–1440, 2006. https://doi.org/10.2514/1.20478.

[128] M. Vasile and P. De Pascale, "Preliminary design of multiple gravity-assist trajectories," *Journal of Spacecraft and Rockets*, vol. 43, no. 4, pp. 794–805, 2006. https://doi.org/10.2514/1.17413.

[129] D. Myatt, V. M. Becerra, S. J. Nasuto, and J. Bishop, "Advanced global optimisation for mission analysis and design," *Final Report. Ariadna id*, vol. 3, p. 4101,

2004.

[130] D. Izzo, "Advances in global optimisation for space trajectory design," in *Proceedings of the International Symposium on Space Technology and Science*, vol. 25, p. 563, 2006.

[131] T. Vinkó and D. Izzo, "Global optimisation heuristics and test problems for preliminary spacecraft trajectory design," *Advanced Concepts Team, ESATR ACT-TNT-MAD-GOHTPPSTD, Sept*, 2008.

[132] J. V. McAdams, D. W. Dunham, R. W. Farquhar, A. H. Taylor, and B. Williams, "Trajectory design and maneuver strategy for the MESSENGER mission to Mercury," *Journal of Spacecraft and Rockets*, vol. 43, no. 5, pp. 1054–1064, 2006. https://doi.org/10.2514/1.18178.

[133] D. G. Yárnoz, R. Jehn, and M. Croon, "Interplanetary navigation along the low-thrust trajectory of BepiColombo," *Acta Astronautica*, vol. 59, no. 1-5, pp. 284–293, 2006. https://doi.org/10.1016/j.actaastro.2006.02.028.

[134] L. Davis, "Bit-climbing, representational bias, and test suit design," in *Proc. Intl. Conf. Genetic Algorithm, 1991*, pp. 18–23, 1991.

[135] P. E. Gill, W. Murray, and M. A. Saunders, "SNOPT: An SQP algorithm for large-scale constrained optimization," *SIAM review*, vol. 47, no. 1, pp. 99–131, 2005. https://doi.org/10.1137/S0036144504446096.

[136] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical programming*, vol. 106, no. 1, pp. 25–57, 2006. https://doi.org/10.1007/s10107-004-0559-y.

[137] R. H. Byrd, J. Nocedal, and R. A. Waltz, "KNITRO: An integrated package for nonlinear optimization," in *Large-scale nonlinear optimization*, pp. 35–59, Springer, 2006. https://doi.org/10.1007/0-387-30065-1_4.

[138] M. Knauer and C. Büskens, "WORHP: Development and Applications of the ESA NLP SOLVER," in *German Success Stories in Industrial Mathematics*, pp. 155–160, Springer, 2021. https://doi.org/10.1007/978-3-030-81455-7_25.

[139] N. Xiong, D. Molina, M. L. Ortiz, and F. Herrera, "A walk into metaheuristics for engineering optimization: principles, methods and recent trends," *International Journal of Computational Intelligence Systems*, vol. 8, no. 4, pp. 606–636, 2015. https://doi.org/10.1080/18756891.2015.1046324.

[140] S. Kirkpatrick, C. D. Gelatt Jr, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983. https://doi.org/10.1126/science.220.4598.671.

[141] F. Glover and M. Laguna, "Tabu search," in *Handbook of Combinatorial Optimization*, pp. 2093–2229, Springer, 1998. https://doi.org/10.1007/978-1-4613-0303-9_33.

[142] H. R. Lourenço, O. C. Martin, and T. Stützle, "Iterated local search," in *Handbook of Metaheuristics*, pp. 320–353, Springer, 2003. https://doi.org/10.1007/0-306-48056-5_11.

[143] E. Goldberg David and H. Holland, "Genetic algorithms and machine learning," *Machine Learning*, vol. 3, no. 2, pp. 95–99, 1988. https://doi.org/10.1023/A:1022602019183.

[144] A. P. Engelbrecht, *Computational intelligence: an introduction*. John Wiley & Sons, 2007.

[145] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95-International Conference on Neural Networks*, vol. 4, pp. 1942–1948, IEEE, 1995. https://doi.org/10.1109/ICNN.1995.488968.

[146] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and*

Human Science, pp. 39–43, Ieee, 1995. https://doi.org/10.1109/MHS.1995.494215.

[147] R. Storn and K. Price, "Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces," *Journal of global optimization*, vol. 11, no. 4, pp. 341–359, 1997. https://doi.org/10.1023/A:1008202821328.

[148] M. Dorigo and L. M. Gambardella, "Ant colony system: a cooperative learning approach to the traveling salesman problem," *IEEE Transactions on evolutionary computation*, vol. 1, no. 1, pp. 53–66, 1997. https://doi.org/10.1109/4235.585892.

[149] J. A. Englander, B. A. Conway, and T. Williams, "Automated mission planning via evolutionary algorithms," *Journal of Guidance, Control, and Dynamics*, vol. 35, no. 6, pp. 1878–1887, 2012. https://doi.org/10.2514/1.54101.

[150] C. M. Chilan and B. A. Conway, "A space mission automaton using hybrid optimal control," in *17th Annual Space Flight Mechanics Meeting*, pp. 259–276, 2007.

[151] B. J. Wall and B. A. Conway, "Genetic algorithms applied to the solution of hybrid optimal control problems in astrodynamics," *Journal of Global Optimization*, vol. 44, no. 4, p. 493, 2009. https://doi.org/10.1007/s10898-008-9352-4.

[152] J. Englander, B. Conway, and T. Williams, "Automated interplanetary trajectory planning," in *AIAA/AAS Astrodynamics Specialist Conference*, p. 4517, 2012. https://doi.org/10.2514/6.2012-4517.

[153] S. Wagner and B. Wie, "Hybrid algorithm for multiple gravity-assist and impulsive delta-V maneuvers," *Journal of Guidance, Control, and Dynamics*, vol. 38, no. 11, pp. 2096–2107, 2015. https://doi.org/10.2514/1.G000874.

[154] N. J. Strange and J. M. Longuski, "Graphical method for gravity-assist trajectory design," *Journal of Spacecraft and Rockets*, vol. 39, no. 1, pp. 9–16, 2002. https:

//doi.org/10.2514/2.3800.

[155] D. Izzo, D. Hennes, L. F. Simões, and M. Märtens, "Designing complex interplanetary trajectories for the global trajectory optimization competitions," in *Space Engineering*, pp. 151–176, Springer, 2016. https://doi.org/10.1007/978-3-319-41508-6_6.

[156] R. Armellin, "Collision avoidance maneuver optimization with a multiple-impulse convex formulation," *Acta Astronautica*, vol. 186, pp. 347–362, 2021. https://doi.org/10.1016/j.actaastro.2021.05.046.

[157] H. Shang and Y. Liu, "Assessing accessibility of main-belt asteroids based on gaussian process regression," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 5, pp. 1144–1154, 2017. https://doi.org/10.2514/1.G000576.

[158] Y. Song and S. Gong, "Solar-sail trajectory design for multiple near-earth asteroid exploration based on deep neural networks," *Aerospace Science and Technology*, vol. 91, pp. 28–40, 2019. https://doi.org/10.1016/j.ast.2019.04.056.

[159] G. Viavattene and M. Ceriotti, "Artificial neural networks for multiple nea rendezvous missions with continuous thrust," *Journal of Spacecraft and Rockets*, vol. 59, no. 2, pp. 574–586, 2022. https://doi.org/10.2514/1.A34799.

[160] R. Matai, S. P. Singh, and M. L. Mittal, "Traveling salesman problem: an overview of applications, formulations, and solution approaches," *Traveling salesman problem, theory and applications*, vol. 1, 2010. https://doi.org/10.5772/12909.

[161] A. Majeed and I. Rauf, "Graph theory: A comprehensive survey about graph theory applications in computer science and social networks," *Inventions*, vol. 5, no. 1, p. 10, 2020. https://doi.org/10.3390/inventions5010010.

[162] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms*. MIT press, 2009.

[163] E. L. Lawler and D. E. Wood, “Branch-and-Bound Methods: A Survey,” *Operations Research*, vol. 14, no. 4, pp. 699–719, 1966. https://doi.org/10.1287/opre.14.4.699.

[164] M. Ceriotti and M. Vasile, “Automated multigravity assist trajectory planning with a modified ant colony algorithm,” *Journal of Aerospace Computing, Information, and Communication*, vol. 7, no. 9, pp. 261–293, 2010. https://doi.org/10.2514/1.48448.

[165] L. F. Simões, D. Izzo, E. Haasdijk, and A. E. Eiben, “Multi-rendezvous spacecraft trajectory optimization with beam P-ACO,” in *European Conference on Evolutionary Computation in Combinatorial Optimization*, pp. 141–156, Springer, 2017. https://doi.org/10.1007/978-3-319-55453-2_10.

[166] M. A. Minovitch, “The invention that opened the solar system to exploration,” *Planetary and Space Science*, vol. 58, no. 6, pp. 885–892, 2010. https://doi.org/10.1016/j.pss.2010.01.008.

[167] S. Campagnola, B. B. Buffington, and A. E. Petropoulos, “Jovian tour design for orbiter and lander missions to Europa,” *Acta Astronautica*, vol. 100, pp. 68–81, 2014. https://doi.org/10.1016/j.actaastro.2014.02.005.

[168] K. W. Kloster, A. E. Petropoulos, and J. M. Longuski, “Europa Orbiter tour design with Io gravity assists,” *Acta Astronautica*, vol. 68, no. 7-8, pp. 931–946, 2011. https://doi.org/10.1016/j.actaastro.2010.08.041.

[169] L. Federici, A. Zavoli, and G. Colasurdo, “Preliminary Capture Trajectory Design for Europa Tomography Probe,” *International Journal of Aerospace Engineering*, vol. 2018, 2018. https://doi.org/10.1155/2018/6890173.

[170] L. A. D’Amario, L. E. Bright, and A. A. Wolf, “Galileo trajectory design,” *Space Science Reviews*, vol. 60, no. 1, pp. 23–78, 1992. https://doi.org/10.1007/BF00216849.

[171] F. Peralta and S. Flanagan, “Cassini interplanetary trajectory design,” *Control Engineering Practice*, vol. 3, no. 11, pp. 1603–1610, 1995. https://doi.org/10.1016/0967-0661(95)00171-P.

[172] N. J. Fox, M. C. Velli, S. D. Bale, R. Decker, A. Driesman, R. A. Howard, J. C. Kasper, J. Kinnison, M. Kusterer, and D. Lario, “The Solar Probe Plus Mission: Humanity’s First Visit to Our Star,” *Space Science Reviews*, vol. 204, no. 1-4, pp. 7–48, 2016. https://doi.org/10.1007/s11214-015-0211-6.

[173] J. Sanchez Perez, W. Martens, and G. Varga, “Solar orbiter 2020 february mission profile,” in *Advances in the Astronautical Sciences*, vol. 167, pp. 1395–1410, 2018.

[174] D. de la Torre Sangrà, E. Fantino, R. Flores, O. C. Lozano, and C. G. Estelrich, “An automatic tree search algorithm for the Tisserand graph,” *Alexandria Engineering Journal*, vol. 60, no. 1, pp. 1027–1041, 2021. https://doi.org/10.1016/j.aej.2020.10.028.

[175] A. F. Heaton, N. J. Strange, J. M. Longuski, and E. P. Bonfiglio, “Automated design of the Europa Orbiter tour,” *Journal of Spacecraft and Rockets*, vol. 39, no. 1, pp. 17–22, 2002. https://doi.org/10.2514/2.3801.

[176] S. Campagnola and R. P. Russell, “Endgame Problem Part 1: V_∞-Leveraging Technique and the Leveraging Graph,” *Journal of Guidance, Control, and Dynamics*, vol. 33, no. 2, pp. 463–475, 2010. https://doi.org/10.2514/1.44258.

[177] N. J. Strange, S. Campagnola, and R. P. Russell, “Leveraging flybys of low mass moons to enable an enceladus orbiter,” *Advances in the Astronautical Sciences*, vol. 135, no. 3, pp. 2207–2225, 2009.

[178] S. Campagnola, N. J. Strange, and R. P. Russell, “A fast tour design method using non-tangent v-infinity leveraging transfer,” *Celestial Mechanics and Dynamical Astronomy*, vol. 108, no. 2, pp. 165–186, 2010. https://doi.org/10.1007/s10569-010-9295-1.

[179] S. Campagnola and R. P. Russell, "Endgame Problem Part 2: Multibody Technique and the Tisserand-Poincare Graph," *Journal of Guidance, Control, and Dynamics*, vol. 33, no. 2, pp. 476–486, 2010. `https://doi.org/10.2514/1.44290`.

[180] V. Maiwald, "Applicability of Tisserand Criterion for Optimization of Gravity-assist Sequences for Low-thrust Missions," in *Proceedings of the International Astronautical Congress, IAC*, 2015.

[181] H. AFSA, *AUTOMATE: Automatic Multi-Gravity Assist Trajectory Design with Tisserand Exploration.* Master thesis, Cranfield University, 2021.

[182] A. Boutonnet and G. Varga, "JUICE—Jupiter icy moons explorer consolidated report on mission analysis (CReMA)," tech. rep., European Space Agency (ESA), 2017.

[183] J. M. Longuski and S. N. Williams, "Automated design of gravity-assist trajectories to Mars and the outer planets," *Celestial Mechanics and Dynamical Astronomy*, vol. 52, no. 3, pp. 207–220, 1991. `https://doi.org/10.1007/BF00048484`.

[184] J. A. Sims, A. J. Staugler, and J. M. Longuski, "Trajectory options to Pluto via gravity assists from Venus, Mars, and Jupiter," *Journal of Spacecraft and Rockets*, vol. 34, no. 3, pp. 347–353, 1997. `https://doi.org/10.2514/2.3215`.

[185] A. E. Petropoulos, J. M. Longuski, and E. P. Bonfiglio, "Trajectories to Jupiter via gravity assists from Venus, Earth, and Mars," *Journal of Spacecraft and Rockets*, vol. 37, no. 6, pp. 776–783, 2000. `https://doi.org/10.2514/2.3650`.

[186] M. R. Patel and J. M. Longuski, "Automated design of Delta-V gravity-assist trajectories for solar system exploration," in *Proceedings of AAS/AIAA Astrodynamics Conference*, 1993.

[187] J. A. Sims, J. M. Longuski, and A. J. Staugler, "V_∞ leveraging for interplanetary missions: Multiple-revolution orbit techniques," *Journal of Guidance, Control, and Dynamics*, vol. 20, no. 3, pp. 409–415, 1997. `https://doi.org/10.2514/2.`

4064.

[188] D. Izzo, V. M. Becerra, D. R. Myatt, S. J. Nasuto, and J. M. Bishop, "Search space pruning and global optimisation of multiple gravity assist spacecraft trajectories," *Journal of Global Optimization*, vol. 38, no. 2, pp. 283–296, 2007. https://doi.org/10.1007/s10898-006-9106-0.

[189] A. Boutonnet, W. Martens, and J. Schoenmaekers, "SOURCE: A Matlab-oriented tool for interplanetary trajectory global optimization. Fundamentals," in *23rd AAS/AIAA Space Flight Mechanics Meeting*, vol. 148, pp. 1449–1468, jan 2013.

[190] M. B. Mora and J. L. Cano, "1st ACT global trajectory optimisation competition: Results found at DEIMOS Space," *Acta Astronautica*, vol. 61, no. 9, pp. 794–805, 2007. https://doi.org/10.1016/j.actaastro.2007.03.007.

[191] S. M. Pessina, S. Campagnola, and M. Vasile, "Preliminary analysis of interplanetary trajectories with aerogravity and gravity assist manoeuvres," in *Proceedings of 54th International Astronautical Congress*, 2003.

[192] A. D. Olds, C. A. Kluever, and M. L. Cupples, "Interplanetary mission design using differential evolution," *Journal of Spacecraft and Rockets*, vol. 44, no. 5, pp. 1060–1070, 2007. https://doi.org/10.2514/1.27242.

[193] M. Vasile and M. Ceriotti, "Incremental techniques for global space trajectory design," *Spacecraft Trajectory Optimization*, vol. 29, pp. 202–237, 2010. https://doi.org/10.1017/CBO9780511778025.009.

[194] A. Gad and O. Abdelkhalik, "Hidden genes genetic algorithm for multi-gravity-assist trajectories optimization," *Journal of Spacecraft and Rockets*, vol. 48, no. 4, pp. 629–641, 2011. https://doi.org/10.2514/1.52642.

[195] O. Abdelkhalik and A. Gad, "Dynamic-size multiple populations genetic algorithm for multigravity-assist trajectory optimization," *Journal of Guidance, Control, and Dynamics*, vol. 35, no. 2, pp. 520–529, 2012. https://doi.org/10.2514/1.

54330.

[196] M. Vasile, "A global approach to optimal space trajectory design," *Advances in the Astronautical Sciences*, vol. 114, pp. 629–647, 2003.

[197] M. Vasile, J. M. R. Martin, L. Masi, E. Minisci, R. Epenoy, V. Martinot, and J. F. Baig, "Incremental planning of multi-gravity assist trajectories," *Acta Astronautica*, vol. 115, pp. 407–421, 2015. https://doi.org/10.1016/j.actaastro.2015.05.033.

[198] K. Deb, N. Padhye, and G. Neema, "Interplanetary trajectory optimization with swing-bys using evolutionary multi-objective optimization," in *International Symposium on Intelligence Computation and Applications*, pp. 26–35, Springer, 2007. https://doi.org/10.1007/978-3-540-74581-5_3.

[199] M. A. Vavrina, J. A. Englander, S. M. Phillips, and K. M. Hughes, "Global multi-objective trajectory optimization with parametric spreading," in *25th AAS/AIAA Astrodynamics Specialist Conference 2017*, 2017.

[200] J. A. Englander, M. A. Vavrina, and A. R. Ghosh, "Multi-objective hybrid optimal control for multiple-flyby low-thrust mission design," in *25th AAS/AIAA Space Flight Mechanics Meeting*, 2015.

[201] A. Ellithy, O. Abdelkhalik, and J. Englander, "Multi-Objective Hidden Genes Genetic Algorithm for Multigravity-Assist Trajectory Optimization," *Journal of Guidance, Control, and Dynamics*, pp. 1–17, 2022. https://doi.org/10.2514/1.G006415.

[202] G. Acciarini, D. Izzo, and E. Mooij, "MHACO: a Multi-Objective Hypervolume-Based Ant Colony Optimizer for Space Trajectory Optimization," in *2020 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1–8, IEEE, 2020. https://doi.org/10.1109/CEC48606.2020.9185694.

[203] F. A. Zotes and M. S. Penas, "Particle swarm optimisation of interplanetary tra-

jectories from Earth to Jupiter and Saturn," *Engineering Applications of Artificial Intelligence*, vol. 25, no. 1, pp. 189–199, 2012. https://doi.org/10.1016/j.engappai.2011.09.005.

[204] M. Vasile and F. Zuiani, "Multi-agent collaborative search: an agent-based memetic multi-objective optimization algorithm applied to space trajectory design," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 225, no. 11, pp. 1211–1227, 2011. https://doi.org/10.1177/0954410011410274.

[205] A. S. Rivkin, R. Anderson, O. Barnouin, N. Chabot, C. Ernst, R. Klima, J. Leary, L. Mehr, H. Seifert, B. A. Cohen, *et al.*, "The main-belt asteroid and NEO tour with imaging and spectroscopy (MANTIS)," in *2016 IEEE Aerospace Conference*, pp. 1–14, IEEE, 2016. https://doi.org/10.1109/AERO.2016.7500757.

[206] ESA, "M5 Call - Technical Annex (ESA-SCI-FESTEC- TN-2016-002)," 2016. Available online at: https://www.cosmos.esa.int/documents/960972/961021/Call+for+M5+Annex.pdf/41250c38-3d9b-44f6-8eaa-5c73b9668237, Last Accessed: February 2022.

[207] H. H. Hsieh and D. Jewitt, "A population of comets in the main asteroid belt," *Science*, vol. 312, no. 5773, pp. 561–563, 2006. https://doi.org/10.1126/science.1125150.

[208] Q.-Z. Ye, P. G. Brown, and P. Pokornỳ, "Dormant comets among the near-Earth object population: a meteor-based survey," *Monthly Notices of the Royal Astronomical Society*, vol. 462, no. 4, pp. 3511–3527, 2016. https://doi.org/10.1093/mnras/stw1846.

[209] B. Gladman, B. G. Marsden, and C. VanLaerhoven, "Nomenclature in the outer Solar System," *The Solar System Beyond Neptune*, vol. 43, pp. 43–57, 2008.

[210] D. Bockelée-Morvan, G. Filacchione, K. Altwegg, E. Bianchi, M. Bizzarro,

J. Blum, L. Bonal, F. Capaccioni, M. Choukroun, and C. Codella, "AMBITION–comet nucleus cryogenic sample return," *Experimental Astronomy*, pp. 1–52, 2021. https://doi.org/10.1007/s10686-021-09770-4.

[211] I. S. Grigoriev and M. P. Zapletin, "GTOC5: Problem statement and notes on solution verification," *Acta Futura*, vol. 8, pp. 9–19, 2014. http://dx.doi.org/10.2420/AF08.2014.9.

[212] N. Ozaki, K. Yanagida, T. Chikazawa, N. Pushparaj, N. Takeishi, and R. Hyodo, "Asteroid Flyby Cycler Trajectory Design Using Deep Neural Networks," *Journal of Guidance, Control, and Dynamics*, pp. 1–16, 2022. https://doi.org/10.2514/1.G006487.

[213] D. Izzo and E. Öztürk, "Real-Time Guidance for Low-Thrust Transfers Using Deep Neural Networks," *Journal of Guidance, Control, and Dynamics*, vol. 44, no. 2, pp. 315–327, 2021. https://doi.org/10.2514/1.G005254.

[214] Y.-h. Zhu and Y.-Z. Luo, "Fast evaluation of low-thrust transfers via multilayer perceptions," *Journal of Guidance, Control, and Dynamics*, vol. 42, no. 12, pp. 2627–2637, 2019. https://doi.org/10.2514/1.G004080.

[215] M. Märtens, D. Izzo, E. Blazquez, M. von Looz, P. Gómez, A. Mergy, G. Acciarini, C. H. Yam, J. H. Ayuso, and Y. Shimane, "The fellowship of the Dyson ring: ACT & Friends' results and methods for GTOC 11," *Acta Astronautica*, 2022. https://doi.org/10.1016/j.actaastro.2022.06.025.

[216] M. Morimoto, H. Yamakawa, M. Yoshikawa, M. Abe, and H. Yano, "Trajectory design of multiple asteroid sample return missions," *Advances in Space Research*, vol. 34, no. 11, pp. 2281–2285, 2004. https://doi.org/10.1016/j.asr.2003.10.055.

[217] Y. Chen, H. Baoyin, and J. Li, "Accessibility of Main-Belt Asteroids via Gravity Assists," *Journal of Guidance, Control, and Dynamics*, vol. 37, no. 2, pp. 623–632,

2014. https://doi.org/10.2514/1.58935.

[218] H. Yang, J. Li, and H. Baoyin, “Low-cost transfer between asteroids with distant orbits using multiple gravity assists,” *Advances in Space Research*, vol. 56, no. 5, pp. 837–847, 2015. https://doi.org/10.1016/j.asr.2015.05.013.

[219] P. Sun, H. Yang, and S. Li, “Accessibility of near-Earth asteroids and main-belt asteroids in a gravity-assisted multi-target mission,” *Planetary and Space Science*, vol. 182, p. 104851, 2020. https://doi.org/10.1016/j.pss.2020.104851.

[220] A. Bellome, J. P. Sanchez Cuartielles, J. Del Ser, S. Kemble, and L. Felicetti, “Efficiency of tree-search like heuristics to solve complex mixed-integer programming problems applied to the design of optimal space trajectories,” *Proceedings of the International Astronautical Congress, IAC*, 2021.

[221] M. Carrillo Barrenechea, *Stochastic Optimisation for Complex Mixed-Integer Programming Problems in Asteroid Tour Missions*. Master thesis, Cranfield University, 2021.

[222] M. Guntsch and M. Middendorf, “Applying population based ACO to dynamic optimization problems,” in *International Workshop on Ant Algorithms*, pp. 111–122, Springer, 2002. https://doi.org/10.1007/3-540-45724-0_10.

[223] M. Guntsch and M. Middendorf, “A population based approach for ACO,” in *Workshops on applications of evolutionary computation*, pp. 72–81, Springer, 2002. https://doi.org/10.1007/3-540-46004-7_8.

[224] M. Guntsch and M. Middendorf, “Solving multi-criteria optimization problems with population-based ACO,” in *International Conference on Evolutionary Multi-Criterion Optimization*, pp. 464–478, Springer, 2003. https://doi.org/10.1007/3-540-36970-8_33.

[225] C. Blum, “Beam-ACO—Hybridizing ant colony optimization with beam search: An application to open shop scheduling,” *Computers & Operations Research*,

vol. 32, no. 6, pp. 1565–1591, 2005. https://doi.org/10.1016/j.cor.2003.11.018.

[226] K. Deb, “Multi-objective Optimisation Using Evolutionary Algorithms: An Introduction,” in *Multi-objective evolutionary optimisation for product design and manufacturing*, pp. 3–34, Springer, 2011. https://doi.org/10.1007/978-0-85729-652-8_1.

[227] K. Vermirovsky, *Algorithms for constraint satisfaction problems*. Master thesis, Masaryk University Brno Faculty of Informatics, 2003.

[228] R. M. Haralick and G. L. Elliott, “Increasing tree search efficiency for constraint satisfaction problems,” *Artificial intelligence*, vol. 14, no. 3, pp. 263–313, 1980. https://doi.org/10.1016/0004-3702(80)90051-X.

[229] P. Van Hentenryck, *Constraint satisfaction in logic programming*. MIT press, 1989.

[230] R. Dechter and D. Frost, “Backjump-based backtracking for constraint satisfaction problems,” *Artificial Intelligence*, vol. 136, no. 2, pp. 147–188, 2002. https://doi.org/10.1016/S0004-3702(02)00120-0.

[231] A. E. Petropoulos, “Problem Description for the 2nd Global Trajectory Optimisation Competition,” 2006. Available online at: https://sophia.estec.esa.int/gtoc_portal/wp-content/uploads/2012/11/gtoc2_problem.pdf, Last accessed: May 2022.

[232] R. Bertrand, R. Epenoy, and B. Meyssignac, “Problem Description for the 4th Global Trajectory Optimisation Competition,” *Centre National d’Etudes Spatiales (CNES)*, 2009. Available online at: https://areeweb.polito.it/gtoc/gtoc4_problem.pdf, Last Accessed: May 2022.

[233] L. Casalino and G. Colasurdo, “Problem Description for the 7th Global Trajectory Optimisation Competition,” 2014. Available online at: https://sophia.estec.esa.int/gtoc_portal/wp-content/uploads/

2014/09/gtoc7_problem_description.pdf, Last accessed: May 2022.

[234] Y.-Z. Luo, Y.-H. Zhu, P. Shu, B. Yan, J.-C. Zhang, H.-X. Shen, A.-Y. Huang, and T.-J. Zhang, "The problem of the 11th global trajectory optimization competition," 2021. Available online at: https://sophia.estec.esa.int/gtoc_portal/wp-content/uploads/2021/12/gtoc11_problem_stmt.pdf, Last accessed May 2022.

[235] Y.-Z. Luo, Y.-H. Zhu, and H.-X. Shen, "The Problem and Result of the 11th Global Trajectory Optimization Competition," *Mechanics in Engineering*, vol. 43, no. 6, p. 1019, 2021. https://doi.org/10.6052/1000-0879-21-508.

[236] H.-X. Shen, Y.-Z. Luo, Y.-H. Zhu, and A.-Y. Huang, "Dyson sphere building: On the design of the GTOC11 problem and summary of the results," *Acta Astronautica*, 2022. https://doi.org/10.1016/j.actaastro.2022.08.040.

[237] L. Casalino, G. Colasurdo, and M. R. Sentinella, "Problem Description for the 3rd Global Trajectory Optimisation Competition," 2007. Available online at: https://sophia.estec.esa.int/gtoc_portal/wp-content/uploads/2012/11/gtoc3_problem.pdf, Last accessed: May 2022.

[238] C. Yam, D. Lorenzo, and D. Izzo, "Low-thrust trajectory design as a constrained global optimization problem," in *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 225, pp. 1243–1251, SAGE Publications Sage UK: London, England, 2011. https://doi.org/10.1177/0954410011401686.

[239] A. Bellome, J.-P. Sánchez, L. Felicetti, and S. Kemble, "Multiobjective design of gravity-assist trajectories via graph transcription and dynamic programming," *Journal of Spacecraft and Rockets*, pp. 1–19, 2023. https://doi.org/10.2514/1.A35472.

[240] E. M. Standish, "Keplerian elements for approximate positions of the major

planets," 2006. Available online at: `https://ssd.jpl.nasa.gov/planets/approx_pos.html`, Last accessed: January 2021.

[241] M. Ceriotti, M. Vasile, and C. Bombardelli, "An incremental algorithm for fast optimisation of multiple gravity assist trajectories," in *58th International Astronautical Congress*, 2007.

[242] J. T. Olympio and D. Izzo, "Designing optimal multi-gravity-assist trajectories with free number of impulses," in *International Symposium on Space Flights Dynamics, ESA-ESTEC, Toulouse, France*, 2009.

[243] F. Tisserand, *Traité de mécanique céleste*. Gauthier-Villars, 1891.

[244] N. J. Strange, *Analytical methods for gravity-assist tour design*. PhD thesis, Purdue University, 2016.

[245] C. S. Arridge, C. B. Agnor, N. André, K. H. Baines, L. N. Fletcher, D. Gautier, M. D. Hofstadter, G. H. Jones, L. Lamy, Y. Langevin, *et al.*, "Uranus Pathfinder: exploring the origins and evolution of Ice Giant planets," *Experimental Astronomy*, vol. 33, no. 2, pp. 753–791, 2012. `https://doi.org/10.1007/s10686-011-9251-4`.

[246] T. A. Pavlak, R. B. Frauenholz, C. E. Helfrich, J. A. Kangas, and J. J. Bordi, "Maneuver design for the juno mission: inner cruise," in *21st AIAA/AAS Astrodynamics Specialist Conference*, p. 4149, 2014.

[247] K. P. Wenzel, R. G. Marsden, D. E. Page, and E. Smith, "The ULYSSES mission," *Astronomy and Astrophysics Supplement Series*, vol. 92, p. 207, 1992.

[248] M. Vasile, E. Minisci, and M. Locatelli, "Analysis of some global optimization algorithms for space trajectory design," *Journal of Spacecraft and Rockets*, vol. 47, no. 2, pp. 334–344, 2010. `https://doi.org/10.2514/1.45742`.

[249] K.-H. Glassmeier, H. Boehnhardt, D. Koschny, E. Kührt, and I. Richter, "The

Rosetta mission: flying towards the origin of the solar system," *Space Science Reviews*, vol. 128, no. 1, pp. 1–21, 2007. https://doi.org/10.1007/s11214-006-9140-8.

[250] S. Wagner, B. Wie, and B. Kaplinger, "Computational solutions to Lambert's problem on modern graphics processing units," *Journal of Guidance, Control, and Dynamics*, vol. 38, no. 7, pp. 1305–1311, 2015. https://doi.org/10.2514/1.G000840.

[251] L. Peacocke, S. Kemble, M. Chapuy, and H. Scheer, "MarcoPolo-R: Mission and Spacecraft Design," in *European Planetary Science Congress*, pp. EPSC2013–116, 2013.

[252] F. Laipert, A. Nicholas, Z. Olikara, R. Woolley, and R. Lock, "Hybrid chemical-electric trajectories for a mars sample return orbiter," *Advances in the Astronautical Sciences*, vol. 168, pp. 2847–2854, 2019.

[253] R. Gallego Fernandez, *Multi Gravity Assist Trajectories Preliminary Analysis with Tisserand Graphs*. Master thesis, Cranfield University, 2016.

[254] S. Li, X. Huang, and B. Yang, "Review of optimization methodologies in global and China trajectory optimization competitions," *Progress in Aerospace Sciences*, vol. 102, pp. 60–75, 2018. https://doi.org/10.1016/j.paerosci.2018.07.004.

[255] C. Bonanno, "An analytical approximation for the MOID and its consequences," *Astronomy and astrophysics*, vol. 360, pp. 411–416, 2000.

[256] S. Milisavljević, "The proximities of asteroids and critical points of the distance function," *Serbian Astronomical Journal*, vol. 180, pp. 91–102, 2010. https://doi.org/10.2298/SAJ1080091M.

[257] J. L. Lagrange, *Mécanique analytique*, vol. 1. Mallet-Bachelier, 1853.

[258] D. d. l. Torre Sangrà and E. Fantino, "Review of lambert's problem," in *ISSFD 2015: 25th International Symposium on Space Flight Dynamics*, pp. 1–15, 2015.

[259] G. Avanzini, "A simple lambert algorithm," *Journal of guidance, control, and dynamics*, vol. 31, no. 6, pp. 1587–1594, 2008. `https://doi.org/10.2514/1.36426`.

[260] C. Wen, Y. Zhao, and P. Shi, "Derivative analysis and algorithm modification of transverse-eccentricity-based lambert problem," *Journal of Guidance, Control, and Dynamics*, vol. 37, no. 4, pp. 1195–1201, 2014. `https://doi.org/10.2514/1.62351`.

[261] R. R. Bate, D. D. Mueller, J. E. White, and W. W. Saylor, *Fundamentals of astrodynamics*. Courier Dover Publications, 2020.

[262] F. Boltz, "Second-order p-iterative solution of the lambert/gauss problem.," *Journal of the Astronautical Sciences*, vol. 32, pp. 475–485, 1984.

[263] J. Garrido and M. Rivo, "A critical review on the state-of-the-art of lambert's problem solvers," in *8th International Conference on Astrodynamics Tools and Techniques*, 2021.

[264] C. T. Russell and C. A. Raymond, *The Dawn Mission to Vesta and Ceres*, pp. 3–23. New York, NY: Springer New York, 2012. `https://doi.org/10.1007/978-1-4614-4903-4_2`.