Journal of Spacecraft and Rockets, Available online 02 April 2023
DOI: 10.2514/1.A35472

Multi-Objective Design of Gravity Assist Trajectories via Graph Transcription and Dynamic Programming

Andrea Bellome *
Cranfield University, Cranfield, MK43 0AL, United Kingdom

Joan-Pau Sánchez †
ISAE-Supaero, Toulouse, 31400, France

Leonard Felicetti ‡ and Stephen Kemble §
Cranfield University, Cranfield, MK43 0AL, United Kingdom

* **Multiple gravity assist (MGA) trajectory design requires the solution of a mixed-integer programming problem, to find the best sequence amongst all possible combinations of candidate planets and dates for spacecraft maneuvers. Current approaches require computing times rising steeply with the number of control parameters, and strongly rely on narrow search spaces. Moreover, the challenging multi-objective optimization needs to be tackled to appropriately inform the mission design with full extent of launch opportunities. This paper describes a methodology based upon a trajectory model to transcribe the mixed-integer space into a discrete graph made by grids of interconnected nodes. The model is based on Lambert arc grids obtained for a range of departure dates and flight times between two planets. A Tisserand-based criterion selects planets to pass-by. Dynamic programming is extended to multi-objective optimization of MGA trajectories and used to explore the graph, guaranteeing Pareto optimality with only moderate computational effort. Robustness is ensured by evaluating the relationship between graph and mixed-integer spaces. Missions to Jupiter and Saturn, alongside challenging comet sample return transfers involving long MGA sequences are discussed. These examples illustrate robustness and efficiency of the proposed approach in capturing globally optimal solutions and wide Pareto fronts on complex search spaces.**

Nomenclature

*PhD Candidate in Aerospace Engineering, School of Aerospace, Transport and Manufacturing; andrea.bellome@cranfield.ac.uk (Corresponding Author).

†Professeur, Department of Aerospace Vehicles Design and Control.

‡Lecturer in Space Engineering, School of Aerospace, Transport and Manufacturing.

§Research Fellow, School of Aerospace, Transport and Manufacturing.

*A version of the manuscript has been presented as: Bellome, A., Sanchez, J. P., Rico Alvarez, J. I., AFSA, H. Kemble, S., and Felicetti, L., An Automatic Process for Sample Return Missions Based on Based on Dynamic Programming Optimization, AIAA SciTech Forum, January 3-7, 2022, San Diego, CA & Virtual. AIAA paper number: 2022-1477. https://doi.org/10.2514/6.2022-1477

$\vec{v}_\infty$ = hyperbolic excess velocity vector relative to a planet, km/s

Δv = manoeuvre impulse, km/s

θ = azimuthal angle of excess velocity at departure, rad

ϕ = elevation angle of excess velocity at departure, rad

k = elevation angle of excess velocity at departure, rad

δ = deflection angle due to planetary swing-by, rad

r_p = swing-by periapsis, km

γ = rotation angle due of planetary swing-by, rad

T = time of flight between two planets, s

α = time fraction at which the maneuver occurs

t = epoch of planetary encounter, MJD2000

N_{rev} = number of revolutions around the Sun

f = cost function

DEF = Defect

DSM = Deep Space Manoeuvre

FE = Full Evaluation

$SODP$ = Single-Objective Dynamic Programming

$MODP$ = Multi-Objective Dynamic Programming

Subscripts

max = maximum

dep = departure

arr = arrival

Superscripts

- = before the fly-by

+ = after the fly-by

I. Introduction

In interplanetary missions, multiple gravity assist (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. This permits gaining or loosing energy with no propellant expenditure, thus allowing exploration of regions of the Solar System that would otherwise be too demanding to reach. For example, Galileo [1], Cassini [2], and the more recent BepiColombo [3], Parker Solar Probe [4], Solar Orbiter [5], and JUICE [6] required or will require multiple fly-bys with

Venus, Earth or even Jupiter to reach the desired scientific orbit.

The design of such missions presents the complication that the trajectory structure, namely the planetary sequence, is not known a priori, but is the objective of the optimization itself, leading to a complex mixed-integer non-linear programming (MINLP) problem [7, 8], also known in literature as a hybrid optimal control problem (HOCP) [9]. 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 coupled optimization problems: (1) the combinatorial part, aiming at choosing the optimal sequence of 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-by parameters and thrust arcs. The MGA problem complexity arises from the fact that these two components are highly coupled. In other words, whilst the quality of a candidate sequence depends highly upon the solution of the continuous optimization, a variation of even a single fly-by body will correspond to a significantly different trajectory path. This in turn requires a different set of continuous variables to be optimised. As such, continuous optimization of the MGA problem is characterized by multiple locally minimum solutions and an optimizable parameter space of complex configuration. Problems that have been tackled in literature to design MGA trajectories can be broadly divided into two main groups: (1) fixed-sequence problem (i.e., assuming the knowledge of the MGA sequence) and (2) variable-sequence problem (i.e., the sequence is not known a priori).

If the MGA sequence is known, a very useful option to find trajectories that visit the selected planets 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, in order to reduce the dimensions of the search space. These are useful for the design of ballistic MGA trajectories when virtually no manoeuvre effort 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 reduce such Δv to a trivially small number. For example, NASA's STOUR programme [10–12] considers the difference between the incoming and outgoing planet-spacecraft relative velocities (in magnitude) at the planetary encounter as a measure of the Δv-cost for a given planet-to-planet phase. ESA's GASP programme [13] employs the so-called powered swing-by model [14] to link successive legs of the overall MGA mission at each planetary encounter, assuming a Δv manoeuvre occurs at the moment of closest approach during a planetary swing-by. GASP solutions are then used to inform successive optimization with either a Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Differential Evolution (DE) or Simulated Annealing (SA). ESA's SOURCE algorithm [15] uses Δv manoeuvres applied immediately after the planetary encounters, allowing the estimation of the cost of a planet-to-planet phase. The main attractive feature of deterministic solution strategies is that the set-up that guarantees global optimum solutions is known a priori. However, the use of approximated Δv manoeuvres at planetary swing-bys does not necessarily correspond to actual Deep Space Manoeuvres (DSMs) in real-world mission scenarios, and the relationship between the different manoeuvre models is not generally

reported. Moreover, the number of routes to be evaluated and stored represents a computational issue, that is amplified if DSMs are considered. Therefore, stochastic metaheuristics are also largely considered to find optimal trajectories for a fixed MGA sequence, among which GAs (e.g., [16, 17]), DE (e.g, [18–20]) and PSO (e.g., [13, 21, 22]) represent valid alternatives. The main advantage of using metaheuristic strategies is that the solution vector does not need a first-guess solution, as they employ an adaptive process to manage the exploration of the search space. However, such approaches are stochastic in nature and thus the set up to consistently converge to optimal solutions is not known a priori and is always problem dependant [18, 21, 23].

To deal with MGA trajectory design without a priori knowledge of the gravity-assist sequence, stochastic metaheuristic strategies are generally employed. Alongside the aforementioned drawbacks of stochastic metaheuristics, they can require quite intense computational effort to handle the mixed-integer complexity of MGA trajectory design. To mitigate such drawbacks, stochastic metaheuristic strategies are usually employed over very small search spaces in terms of launch dates, transfer times and DSM locations. Due to the mixed-integer nature of the problem at hand, an approach based on nested loop optimization was presented in [24–27]. In nested optimization, an integer GA is used on the outer loop to search for the optimal MGA sequence. The quality of such sequences is assessed in the inner loop by looking for single-objective optimal solutions, employing continuous design variables and using a combination of PSO and DE. Computational effort rises steeply (to reach the order of multiple days of parallel computing) with the dimensions of the search space. A similar approach is employed in [28], where a gradient-based single-objective optimization is used for the inner loop strategy, but with no guarantee of consistent convergence to optimal solutions. This is a typical issue of metaheuristic strategies. Ant Colony Optimization (ACO) has also been employed [29], to construct MGA sequences that are Δv-optimal, by exploiting DSMs at the apses of planet-to-planet transfer arcs. This assumes knowledge of the departing date and use of a simplified dynamical model for planetary orbits. Strategies based on hidden-genes GA have also been employed in [30, 31] and used to find minimum Δv solution for MGA trajectories, assuming that the launch happens anytime within a window spanning a 30-days range. A hybridization between incremental tree-graph exploration and a bio-inspired probabilistic algorithm has also been proposed in [32], in order to find optimal solutions with respect to propellant consumption for missions to asteroids, Jupiter and Mercury.

Most of the above optimization methods, either on fixed-sequence or variable-sequence problems, used single-objective optimization, minimizing the overall Δv consumption. However, beyond the challenge of finding the global optimum of a complex MINLP/HOCP, practical mission feasibility studies for MGA trajectory design also require an accurate 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 multi-objective optimization problem, in which multiple objectives that are competing with each other in the mission design need to be optimized simultaneously.

For the multi-objective formulation of the MGA problem, primarily stochastic metaheuristics have been employed. A non-dominated-sorting GA (NSGA-II) has been employed in [33], assuming a maximum of three swing-bys and no

DSMs during the transfer. NSGA-II has also been used in [34], in conjunction with parametric spreading. It is also used as the outer loop optimizer in [35] with monotonic basin hopping and a sparse nonlinear optimizer as inner loop optimizers. This results in quite intense computational effort, spread over massive parallel computations. The concept of a hidden-genes GA for multi-objective optimization is used in [36], constraining the bounds of the launch date to a one-month range and assuming a maximum of three fly-bys for missions to Mercury and Jupiter, and a maximum of four fly-bys for missions to Saturn. A multi-objective variant of both ACO [37] and PSO [38] has been employed to optimize MGA transfers with the knowledge of the planetary sequence. An agent-based mimetic algorithm has also been introduced [39] for multi-objective optimization assuming a priori the MGA sequence used to reach Saturn. This also considers no DSMs during the transfer.

From the discussion above, there is a gap in literature regarding the exploitation of the advantages of deterministic strategies to handle the multi-objective optimization of MGA trajectories in a robust and efficient manner. This gap includes the need to mitigate the main issues of metaheuristic strategies, primarily related to the dimensions of the search space, the implied computing time and the required a priori knowledge of the trajectory sequence or DSMs positions. Therefore, in order to address this gap, the novelties introduced in this paper are the following:

- Multi-objective optimization of MGA transfers with DSMs is performed in a deterministic manner. The MGA trajectory design problem is transcribed into a graph of interconnected nodes that are linked by an approximated Δv occurring at each planetary swing-by. To mitigate one of the issues related to such an approach (i.e., the high number of routes to be considered), dynamic programming principles [40] are extended to these transfers in order 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 the Tisserand parameter [41] is employed during the search for reachable planets that may be used to aid the transfer.
- An analytic procedure based on approximated Δv (i.e. velocity defect) removal and employing position constraints is used to assess the relationship between manoeuvre model types. This is used to address 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.
- Very large search spaces are considered in the optimization of long MGA sequences, in the context of a test case based on a comet sample return mission scenario.

The whole procedure provides wide Pareto fronts for the missions of interest whilst significantly reducing the computational effort needed to find solutions. It also identifies novel transfers with respect to current literature for missions to Jupiter and Saturn. Therefore the main disadvantages of deterministic and metaheuristic strategies are thus mitigated because (1) dynamic programming guarantees, by means of a computationally efficient process, global Pareto

optimality on the transcribed space; (2) the relationship between thrust model types is demonstrated.

The paper is structured as follows: section II introduces the problem and the specific approach followed; section III describes the method of generating feasible MGA sequences by means of the Tisserand parameter; section IV discusses the graph transcription employed and the specific graph structure of the search space; section V describes dynamic programming as an efficient way to deal with the exploration of possible transfer strategies in the transcribed space; section VI discusses the relationship between the transcribed and the un-transcribed space, presenting an analytical procedure to estimate the effect of adding DSMs between two planets; finally, section VII presents numerical test cases useful to demonstrate the efficiency and robustness of proposed methodologies, when applied to complex mission scenarios involving large search spaces and long MGA sequences.

II. Problem Definition

The MGA trajectory design is a global optimization problem in its nature. For a given planetary sequence, several locally optimal trajectories exist in terms of planet encounters, manoeuvres and fly-by parameters. The objective function exhibits a complex dependency on the control parameters. 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 solving a MINLP/HOCP problem. In such problems, there is a function $F(x, y)$ to be minimized, depending on both integer (x) and continuous-varying variables (y). The function $F(x, y)$ can either include one single objective or a number n_{obj} of competing objectives to be optimized simultaneously. A general MINLP contains the following structure:

$$\begin{aligned}
\text{Minimize: } & F(x,y) = f_1(x,y), \ldots, f_k(x,y),\ k = 1, \ldots, n_{obj} \\
& \text{where } x \in \mathbb{Z}^{n_{int}}, y \in \mathbb{R}^{n_{cont}}, n_{cont}, n_{int} \in \mathbb{N} \\
\text{Subject to: } & g_i(x,y) \leq 0, i = 1, \ldots, m \in \mathbb{N} \\
& x_{lb} \leq x \leq x_{ub},\ x_{lb}, x_{ub} \in \mathbb{Z}^{n_{int}} \\
& y_{lb} \leq y \leq y_{ub},\ y_{lb}, y_{ub} \in \mathbb{R}^{n_{cont}}
\end{aligned} \tag{1}$$

where $g_i(x, y)$ represents the constraints of the problem at hand (e.g., overall mission duration or Δv) and m is their number; (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 included 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 included in the continuous-varying vector y. The functions $F(x, y)$ and $g_i(x, y)$ would then represent critical mission parameters such as the Δv cost of the entire transfer and the mission duration, as well as other mission-specific objectives.

Figure 1 shows an example trajectory which follows an Earth (E) – Venus (V) – Mars (M) sequence (EVM) with

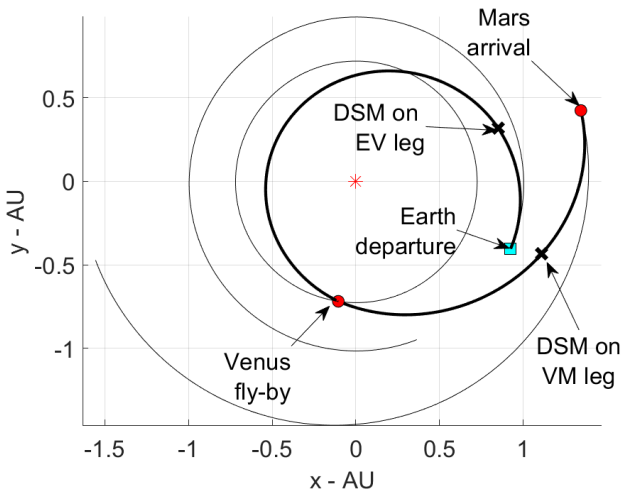


Fig. 1 Example of an EVM trajectory with DSMs on both EV and VM legs.

DSMs on both EV and EM legs. Table 1 provides a description of the integer and continuous variables involved in the problem at hand. In this example, vector x includes a total of 3 objects (i.e., the planets), and vector y includes 10 variables, defining all of the events necessary to characterize the trajectories followed by the spacecraft between each planet. The model used here is the so-called MGA-DSM [21, 42, 43], on which a DSM is assumed to occur between two consecutive planetary encounters. In this model, a propagated arc in two-body dynamics (i.e. Keplerian arc) is assumed after each object encounter (i.e. for the departing and swing-by planets) until the DSM epoch is reached. This is followed by another Lambert's arc between the DSM location and the next planetary encounter.

One should notice that the number of optimizable parameters could rise sharply if either more complex transfer options or more complex dynamical frameworks are implemented, such as low thrust manoeuvres and multi-gravity field models. Other strategies are also found in literature to model the transfers between two planetary encounters, such as multiple-shooting algorithms [44]. Furthermore, primer vector theory [45] may be used to mitigate some of the difficulties associated with the MGA-DSM model, by defining the nature and number of manoeuvres needed between any two planetary encounters. However, for the purposes of the present paper, the MGA-DSM model is deemed sufficient to provide representative trajectories of the MGA problem.

In this study, we consider $F(x, y)$ being a function of two objectives to be optimized, i.e., $F(x, y) = (f_1(x, y), f_2(x, y))$, for which:

$$\begin{cases} f_1 &= v_{\infty,dep} + \sum_{i=1}^{n_{int}-1} \Delta v_i + v_{\infty,arr} \\ f_2 &= \sum_{i=1}^{n_{int}-1} T_i \end{cases} \tag{2}$$

where $v_{\infty,dep}$ and $v_{\infty,arr}$ are the spacecraft velocities relative to the departing and arrival body, respectively, and Δv_i

Table 1 Integer and continuous variables for the MINLP instance of the MGA problem with DSMs.

Integer variables (x)	**Description**
$x_i, \forall i = 1, ..., n_{int}$	Objects in the sequence (n_{int} is the number of integer variables)

Continuous variables (y)	**Description**
$[t_0, v_{\infty,dep}, \theta, k, T, \alpha]_1$	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; (θ, k) define the heliocentric direction of the spacecraft launch as per [42]; T is the transfer time between two bodies; α is the time fraction at which a DSM is performed.
$[r_p, \gamma, T, \alpha]_i, \forall i = 2, ..., n_{int} - 1$	For all the successive planet-to-planet legs: r_p and γ are periapsis and inclination of the fly-by hyperbola, respectively.

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. (1) and (2), with no a priori knowledge of the problem, would only be feasible for formulations with very small search domains for both integer variables x and the continuously-varying vector y. This implies 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 in order to manage this complexity efficiently. To do so, the following pipeline is used, briefly introduced here, and expanded in successive sections:

- A criterion to select successive planetary encounters is used (section III). This is mainly to assess the feasibility of different sequences in designing the mission under consideration, without wasting effort in un-promising areas of the search space.
- A transcription of the problem from a mixed-integer formulation to a discrete optimization is useful (section IV) to explore the search space in an efficient manner. This involves exploitation of the properties of the sub-structure of the problem and allows suitable graph-traversing techniques to be applicable (section V).
- A refinement step is finally implemented (section VI) which takes each qualifying solution identified in the previous steps and finds all of the relevant parameters of the original problem.

III. Generation of MGA Sequences via a Tisserand Graph

The first step of the solution of MGA trajectory design, i.e., the combinatorial optimization of feasible sequences, usually appears within the logic of multi-fidelity processes [46]. Within this paradigm, Tisserand graphs [41] are often used as the lowest fidelity approximation. These graphs represent a valid option to quickly assess the feasibility of different sequences connecting two celestial objects.

A Tisserand graph is a tool which makes use of energetic consideration to build different gravity-assist sequences

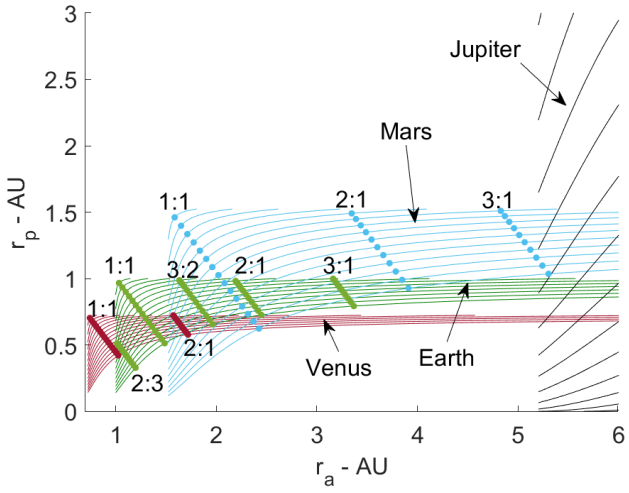


Fig. 2 Orbits at different v_∞ at Venus, Earth, Mars, and Jupiter. Some resonant orbits are also represented for Venus, Earth, and Mars.

[41, 47]. It has been used over the past decades for trajectory design of interplanetary missions. Some examples include the Galileo Orbiter's trajectory design [48], as well as Europa Orbiter design [49, 50]. The Tisserand graph has also been employed in building the winning trajectory of the 6^{th} edition of the Global Trajectory Optimization Competition (GTOC) [51]. Several modified versions were developed to adapt it to specific purposes, as in the Saturn moon tours [52], the case of the circular-restricted 3-body problem [53], and low-thrust propulsion trajectories [54].

The Tisserand graph can be obtained by parametrizing the spacecraft orbit with respect to the velocity relative to the gravity-assist body, also called infinity velocity $\vec{v}_\infty$ (see [41, 47] for more details). Figure 2 represents a Tisserand graph where each point corresponds to a spacecraft orbit, in terms of apoapsis r_a and periapsis r_p, that crosses Solar System planets (Venus to Jupiter in this case) at different $\vec{v}_\infty$. Intersections between lines represent possible transfer orbits between two planets, i.e., orbits that cross two planets at the same time. Figure 2 also highlights different resonant orbits at Venus, Earth, and Mars. These are useful to connect two different planets when the change of the orbit induced by a single fly-by is not sufficient (usually due to the minimum altitude above the planet that the spacecraft can have during the gravity-assisted manoeuvre). Non-resonant ballistic trajectories can also be used to bridge the gap between different contours, especially if small bodies are involved in the transfer (e.g., Saturn moons) [55]. However, transfer orbits on Tisserand graph only exist from an energetic point of view, since the graph contains no explicit information regarding the planetary phase and transfer time, and any approximation employed should be taken with care as it might not correctly map into higher fidelity models [41, 56]. Apart from planetary sequences, information on infinity velocities needed to reach specific orbits can also be sought from Tisserand graphs, and used as non-linear constraints to reduce the search space of successive optimization processes [15].

Therefore, exploring a Tisserand graph consists in the evaluation of the effect of all the possible sequences of

planetary swing-bys as modelled by the parameters of the Tisserand invariant. In this way, it is possible to automatically enumerate all the planetary sequences and excess velocities that are feasible to reach the desired target orbit, by connecting different contours when intersections occur. This is performed by a Breadth First (BF) search strategy [57], and the full algorithmic implementation is described in [56, 58, 59], similarly to [41, 47]. The number of combinations to explore is usually manageable by complete enumeration strategies like BF, since a maximum of 4-5 swing-bys is sufficient to reach any object in the Solar System. It also generally results in transfer times from 5 to more than 20 years (see section VII). Nevertheless, one can envisage applications such as Jupiter and Saturn moon tours [48, 49, 52] in which longer sequences might be needed, depending on specific mission-related constraints. Therefore, incomplete enumeration processes [60] may be relevant.

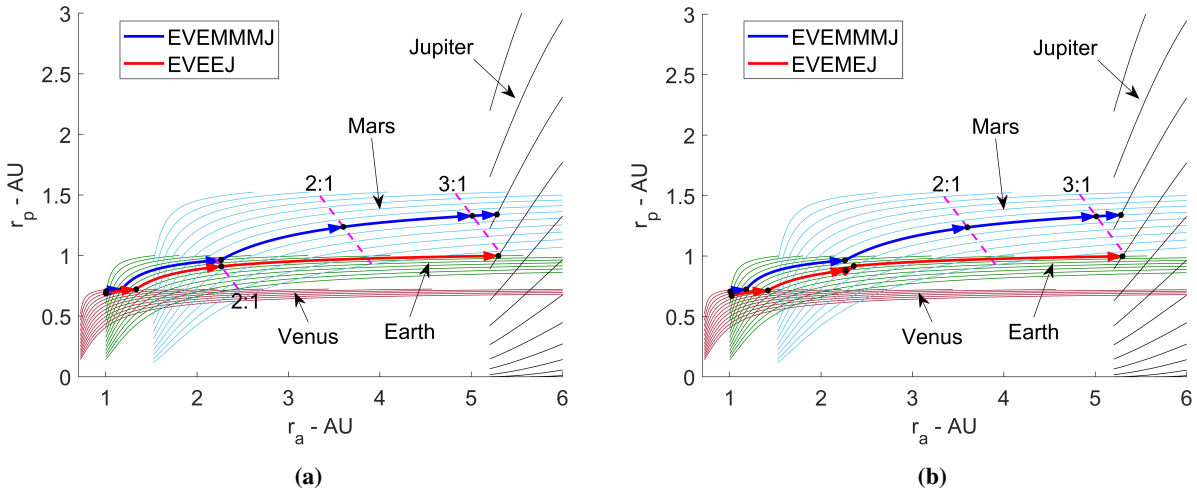


Fig. 3 EVEMMMJ sequence compared to EVEEJ (a) and EVEMEJ (b).

Tisserand graph exploration is thus very useful for obtaining a wide range of planetary encounters that are feasible from an energetic point of view, and one might recognize transfers that approximate global optimality for a given transfer scenario. A good application is the Earth (E) – Jupiter (J) scenario, where an automatic exploration of the Tisserand graph finds 27 sequences with 3 to 5 fly-bys for an infinity velocity range of 3 to 5 km/s at the Earth and as low as 5 km/s at Jupiter with step sizes of 1 km/s between different contour lines. Few seconds of computational time are needed to scan the Tisserand graph on a standard laptop (i.e., 4 GHz single core). Despite the relatively high number of trajectory options, the convenient search space transcription (section IV) in conjunction with dynamic programming approach (section V) leads to manageable computational effort (section VII). A robust and efficient exploration of the whole search space can be performed without a priori knowledge on the solution or the need for stochastic metaheuristic strategies. It should be noted that while an explicit exploration of integer search space is described here (via Tisserand graph), the process can also be implicitly implemented simultaneously with the mixed-integer search of MGA sequences (as discussed in section V), with no conceptual or computational difference.

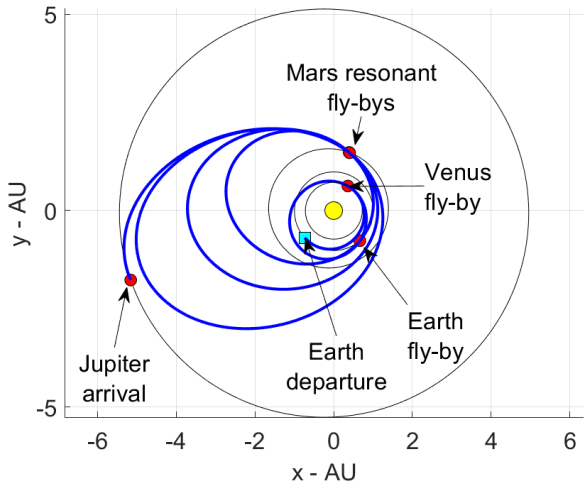


Fig. 4 EVEMMMJ transfer exploiting 2:1 and 3:1 resonant transfers on the successive MM legs.

Ultimately, Tisserand graph exploration provides a quick process to explore the integer search space and, as demonstrated by Figure 3, captures efficiently all notable solutions. Figure 3 depicts only 3 solutions out of the 27 sequences identified, including the well-known EVEEJ (Figure 3.a) and the EVEMEJ (Figure 3.b) considered, for example, for Galileo mission [1] or JUICE [6, 61]. Moreover, other very notable solutions are also identified, such as an EVEMMMJ with 2:1 and 3:1 resonant ratio on each consecutive MM leg. This well-approximates the theoretical global optimum resulting from a fully ballistic transfer with lowest possible $v_{\infty,dep}$ and $v_{\infty,arr}$ (i.e., half ellipse transfer from Earth to Venus on the first leg and half ellipse from Mars to Jupiter on the last leg). Figure 4 depicts such a transfer, which is identified automatically by the multi-objective MGA design described in section IV.

Thus, all the feasible sequences that are identified by the Tisserand map complete exploration are stored and used in successive steps (section IV) to find trajectories between planets.

IV. Optimization of MGA Transfers

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. Compared to existing literature, the efficiency of the whole process lies on this second step, i.e., 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, in which the optimization of a given sequence is performed sequentially, i.e., one planet-to-planet leg at a time. This section deals with this transcription process, while section V describes the MODP as a method to explore the transcribed search space.

A. Search Space Transcription

The idea is to consider for any given planet-to-planet leg a DSM occurring immediately after each planetary encounter, i.e., $\alpha_i \forall i = 1, ..., n_{int} - 1$ from Table 1. 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 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:

$$\Delta v = \begin{cases} ||\vec{v}_\infty^+| - |\vec{v}_\infty^-|| & \text{if } \delta \leq \delta_{max} \\ \sqrt{|\vec{v}_\infty^+|^2 + |\vec{v}_\infty^-|^2 - 2|\vec{v}_\infty^+||\vec{v}_\infty^-|\cos(\delta_{max} - \delta)} & \text{otherwise} \end{cases} \tag{3}$$

where $\vec{v}_\infty^-$ and $\vec{v}_\infty^+$ are 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 degree-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 reads as:

$$\delta_{max} = 2 \arcsin\left(\left(1 + \frac{r_{p,min}|\vec{v}_\infty^-|^2}{\mu_{pl}}\right)^{-1}\right) \tag{4}$$

with $r_{p,min}$ being the minimum periapsis of the fly-by hyperbola as in [41] and μ_{pl} the gravitational constant of the fly-by planet. In other words, if the required deflection $\delta \leq \delta_{max}$, the vectors $\vec{v}_\infty^-$ and $\vec{v}_\infty^+$ can be aligned with the fly-by, while if $\delta > \delta_{max}$, then a direction change needs to be incorporated at greater Δv cost.

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. Thus, the overall Δv consumption accumulated along the 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. (1) and Table 1 only includes the departing date from the first object (t_1) and the transfer times between successive objects (T_i), i.e. $y = [t_1, T_1, \ldots, T_{n_{int}-1}]$. This step reduces the number of decision variables in the continuous domain.

The main advantage of modelling the problem using defects is that each planet-to-planet leg depends only upon the previously visited object through the vector $\vec{v}_\infty^-$ (see section IV.B). To usefully exploit this property, the decision variables are applied over grids of departing dates and transfer times between two successive planetary encounters. 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. (2)). As an 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:

- Lambert problems are solved over a grid of departure dates and transfer durations for the first leg, i.e., a, b.
- At the start of the next leg, namely b, c, departure dates are updated using the arrival epochs at b from the previous leg, plus the range of duration for the current leg towards c. This step is repeated for all consecutive legs.
- For each fly-by in the sequence, for all the incoming routes to a planet, that correspond to the different departure epoch and transfer times from the previous leg, the incoming relative velocity $\vec{v}_\infty^-$ is compared with the outgoing relative velocity $\vec{v}_\infty^+$ for routes to the next flyby (for matching arrival and departure times) and the defect is computed as from Eq. (3).

Tisserand-informed exploration of planetary fly-bys as from section III also identifies 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 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 [28]. Thus, if the Tisserand exploration has identified a resonant leg within an MGA sequence, this is not generated by standard Lambert solvers, which give issues for resonant transfers. For a given approach vector $\vec{v}_\infty^-$, an infinite number of post fly-by resonant solutions exist, for the defined resonance. They 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.

B. Graph Structure of the Search Space

Discrete problems are usually modelled with a search space consisting of 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. The nodes are connected by paths of a fixed length. As an optimization problem, the shortest path or tour around all cities is sought. The TSP example may be translated 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 section IV.A, the cost of the path connecting two objects is associated with the Lambert arc connecting both objects at their respective encounter epochs. Figure 5 illustrates the spacecraft trajectory between two points P_j and P_k. P_j and P_k each have an associated encounter epoch $t_{(\cdot)}^{enc}$. Therefore the time of flight between the two is uniquely defined (i.e., $T = t_k^{enc} - t_j^{enc}$) 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 in Eq. (3). Consequently, the cost of a given leg is not unique

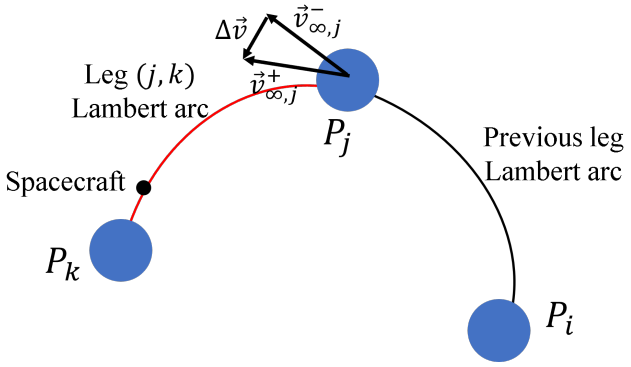


Fig. 5 Sketch of spacecraft trajectory and Δv between point P_j and P_k.The spacecraft is moving from point P_j and P_k on the red track, having visited previously object P_i.

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 also needs to consider the previously visited one, say P_i, so that for the triplet (P_i, P_j, P_k), there exists a unique cost. One should note that this optimal sub-structure property in the form of a triplet of individual nodes is common to all problems where fly-bys are to be considered (e.g., [62]).

Because of this sub-structure of unique triplets, the search space can be modelled as a graph made by interconnected nodes. Each node includes a couple of points with their encounter epochs. Following the example of section IV.A, 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 $[A, B, C]$:

- $A = (at_1, bt_2)$
- $B = (bt_2, ct_3)$
- $C = (ct_3, dt_4)$

Consequently each node includes a trajectory linking two consecutive objects being visited at the specified epochs. When connecting two consecutive nodes, the first object in one node must be equal to the second object included in the previous node. The cost of each connection between two nodes is then given by the Δv in Eq. (3). 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 now 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, allowing dynamic programming techniques to be applicable to the problem at hand.

V. Graph Exploration via Dynamic Programming

Although the transcription process described in section IV.A is similar in related works [10–13, 15], in the present paper the sub-structure of unique triplets as described in section IV.B is usefully exploited to obtain globally optimal transfers in the transcribed search space via dynamic programming principles. Therefore the discrete problem of MGA

missions is conveniently modelled as a tree-graph. Each node of 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, in 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-levels of the tree-graph. This also allows to better handle the constraints from Eq. (1), 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.

Amongst algorithms usually employed to scan tree-graphs, Depth First (DF) or Breadth First (BF) strategies are the most common [57]. These are known to be complete strategies, i.e., they allow identification of the global optimum in discrete/combinatorial problems by keeping all of the possible branched nodes whilst 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 [60], since the computational effort is bounded by heuristics that prevent the exploration of non-promising branches, so that only a limited number of nodes are kept at the selection step. For this reason, BS sacrifices the guarantee of global optimality in favour of computational efficiency.

A. Single- and Multi-Objective Dynamic Programming

Dynamic programming mitigates the computational burden associated with 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 decisions of the sequences 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 [40].

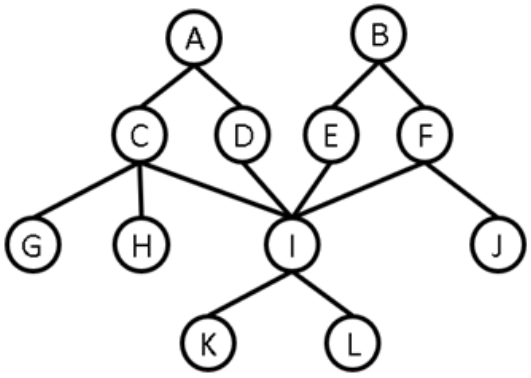


Fig. 6 Tree-graph with common node at the third depth-level.

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 6 shows a representation of this principle. One notices that, at the third depth-level, the tree expansion reaches node I from four different paths, namely: ACI, ADI, BEI, and BFI (one should recall that each node is made by a couple of objects and their visiting epochs as from section IV.B). Assume now that the cost of sequence BFI (e.g., f_1 or f_2 as from Eq. (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. Hence ACI, ADI and BEI need not be kept for further consideration, since they will always be worse than 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 identification, in an automatic manner, the sequence that minimizes a specific objective with the lowest number of paths to be stored in memory.

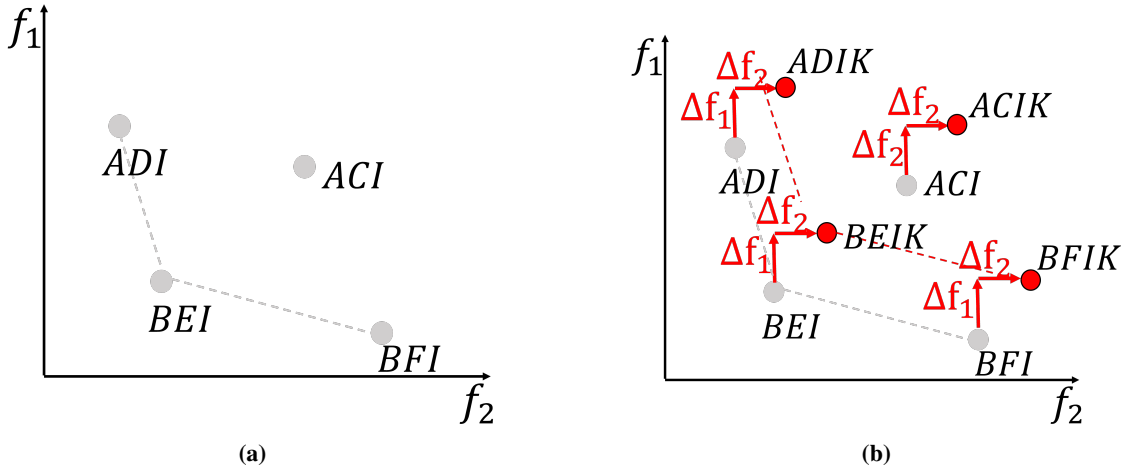


Fig. 7 Representation of different paths arriving to the same node at a specific tree-depth in the f_1, f_2 plane (a) and effect of adding a node to the same sequences (b). Dotted lines link nodes on the Pareto front.

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 [63]. Analogously to the SODP case, if multiple sequences arrive at the same node at a given tree-depth, then only the paths belonging to the Pareto front are useful and so need to be kept for further consideration.

This can be seen intuitively following the same example of Figure 6. 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 with objective functions $F(x, y) = f_1(x, y), ..., f_k(x, y)$ as main axes. For the sake of simplicity, let's consider just two objectives f_1 and f_2 (e.g.,

as from Eq. (2)). The representation of the nodes in such space is given in Figure 7.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 7.b, which is the same for all the sequences. Therefore the Pareto front is 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 so is not needed for further expansion. Therefore in an analogous way to the SODP case, in discrete optimization problems, multi-objective dynamic programming (MODP) allows identification of the optimal Pareto front with the lowest number of paths to be stored in memory in an automatic manner.

When expanding the tree-graph, one can either: (1) generate the list of feasible planetary sequence by means of Tisserand-based information as from section III 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 paper, the tree-graph expansion branches new trajectory legs only if the sequence is within a pre-loaded list of sequences (which is the result of section III). One also notices that since the multi-objective optimization is performed on the transcribed space, it is important to keep all the different planetary sequences that arrive at the target planet, to then assess the leveraging in the refinement step for a correct trade off analysis. This is because different planetary sequences can perform in different ways in the refinement step (see also section VII.D). In other words, if two different planetary sequences arrive to the same node, one should keep them both.

VI. Refinement of Solutions with Mid-course DSMs

To sum up, by applying the transcription process described in section IV.A, 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 aimed at optimizing the continuous design variables given a fixed sequence and visiting epochs. 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).

The 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 preceding 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 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 $\vec{C} = [\vec{r}_{rel}, \Delta v_\infty]^T$, for which $\vec{r}_{rel} = \vec{r} - \vec{r}_{pl}$ 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 at t_f. The control variables are thus $\vec{U} = [\Delta\vec{v}(t_0), t_f]^T$, on which $\Delta\vec{v}(t_0) = [DSM_v, DSM_p, DSM_n]^T$ is the manoeuvre vector, written in the reference frame identified by $\hat{v}_v$, $\hat{v}_p$ and $\hat{v}_n$ as define above.

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

$$\frac{\partial\vec{C}}{\partial\vec{U}}\Delta\vec{U} = \begin{bmatrix} 0 & 0 & 0 & -\Delta v_\infty \end{bmatrix}^T \tag{5}$$

via inversion of $\frac{\partial\vec{C}}{\partial\vec{U}}$. From Eq. (5), 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:

$$\frac{\partial\vec{C}}{\partial\vec{U}} = \begin{bmatrix} \frac{\partial\vec{r}_{rel}(t_f)}{\partial\Delta\vec{v}(t_0)} & \frac{\partial\vec{r}_{rel}(t_f)}{\partial t_f} \\ \frac{\partial|\vec{v}_{rel}|(t_f)}{\partial\Delta\vec{v}(t_0)} & 0 \end{bmatrix} \tag{6}$$

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 the spacecraft and planet velocities at t_f, respectively). In Eq. (6), the term $\frac{\partial|\vec{v}_{rel}|(t_f)}{\partial t_f} = 0$ because the acceleration due to the Sun's gravity is the same for both spacecraft and planet (see also Appendix for further details).

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

$$\Phi = \begin{bmatrix} \frac{\partial\vec{r}(t_f)}{\partial\vec{r}(t_0)} & \frac{\partial\vec{r}(t_f)}{\partial\vec{v}(t_0)} \\ \frac{\partial\vec{v}(t_f)}{\partial\vec{r}(t_0)} & \frac{\partial\vec{v}(t_f)}{\partial\vec{v}(t_0)} \end{bmatrix} \tag{7}$$

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:

$$\frac{\partial\vec{r}_{rel}(t_f)}{\partial\Delta\vec{v}(t_0)} = \begin{bmatrix} \frac{\partial\vec{r}(t_f)}{\partial\vec{v}(t_0)}\hat{v}_v & \frac{\partial\vec{r}(t_f)}{\partial\vec{v}(t_0)}\hat{v}_p & \frac{\partial\vec{r}(t_f)}{\partial\vec{v}(t_0)}\hat{v}_n \end{bmatrix} \tag{8}$$

Moreover, one has:

$$\frac{\partial\vec{r}_{rel}(t_f)}{\partial t_f} = \vec{v}(t_f) - \vec{v}_{pl}(t_f) \tag{9}$$

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:

$$\frac{\partial|\vec{v}_{rel}|(t_f)}{\partial\Delta\vec{v}(t_0)} = \left[\frac{\partial|\vec{v}_{rel}|(t_f)}{\partial DSM_v} \quad \frac{\partial|\vec{v}_{rel}|(t_f)}{\partial DSM_p} \quad \frac{\partial|\vec{v}_{rel}|(t_f)}{\partial DSM_n}\right] \tag{10}$$

one uses:

$$\frac{\partial|\vec{v}_{rel}|(t_f)}{\partial DSM_i} = \frac{\partial(\vec{v}(t_f) - \vec{v}_{pl}(t_f))}{\partial DSM_i} \cdot \hat{v}_{rel} = \frac{\partial\vec{v}(t_f)}{\partial DSM_i} \cdot \hat{v}_{rel} \tag{11}$$

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. (6) using Eq. (8), (9) ans (10), and thus $\Delta\vec{U}$ from Eq. (5). 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, i.e., repeating the evaluation over a sequence of values of t_0 to assess the efficiency of adding a DSM to remove a successive defect, as done in section VII.C. 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 mid-course 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. (8) and (11) are obtained from the standard STM as in Eq. (7) 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 mid-course 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) to (11). 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 full solutions, i.e., solving 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. The refinement process in the MGA-DSM model is carried by re-optimizing the solutions using a PSO. To reconstruct MGA-DSM model parameters from defects model the definition of launch angles (θ, ϕ) and fly-by parameters (r_p, γ) described in [21] is used. 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 and fly-by parameters on a ± 15 % range. Typically, few seconds are needed for the PSO to converge (approximately 40 iterations with 1200 particles on 4 GHz laptop).

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 8, representing on the (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

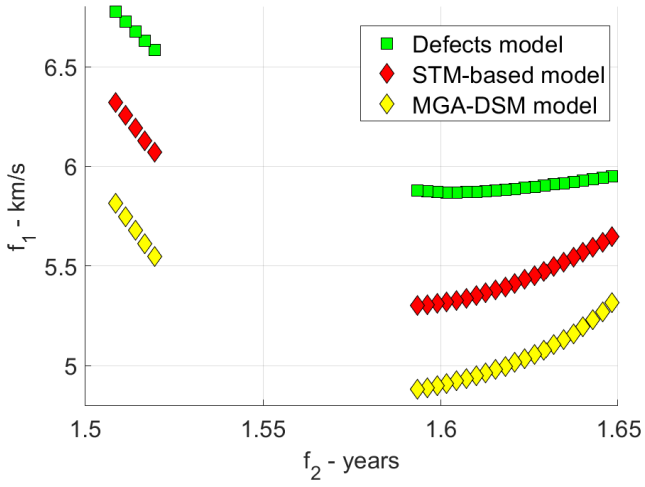


Fig. 8 f_1 and f_2 values for different EVV trajectories arriving at the node (V,t_1,V,t_2) computed with models of different fidelity.

case) and their encounter epochs (t_1 = -589 MJD2000 and t_2 = -183 MJD2000 in this example). The maximum defect allowable is 2 km/s (see also later section VII.B, which explains the roughly one-month gap in Figure 8. One appreciates that all the solutions incoming to a given node belong to a single-funnel structure [23]. Therefore, the SODP/MODP selection process as from section V 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 8 also provides an illustration of the potential value obtained from using STM-based predictions to the MGA-DSM model. 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 described by Eq. (5) to (11) while in the branching step as from section V could be beneficial, because DSMs lie closer to the fully optimized solutions. The price is a 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. It is worth noticing that one might consider the use of a manoeuvre at fly-by periapsis at a lower Δv cost, and thus a better f_1 estimate (e.g., as in [13]). For the present pipeline, this might be considered at the refinement stage, rather than at the exploration step, since such an implementation would scale down the Δv costs, not altering significantly the topography of the search space. Therefore, a simple Δv as in Eq. (3) is deemed sufficient to provide accurate results for the application considered.

VII. Numerical Results and Discussion

Applications of the pipeline presented in sections III, IV, V and VI are here discussed. These follow a trend of increased complexity and are briefly introduced here:

- A transfer towards Saturn is optimized with respect to the two objectives f_1 and f_2 as in Eq. (2), assuming the sequence from Earth to Saturn to be known, i.e., EVVEJS. This is like the Cassini design problem proposed by ESA [†]. However, compared to most of the literature on the same problem, the multi-objective optimization is tackled here. 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 VII.A) and to test the efficiency of using the transcription process alongside SODP/MODP approaches when exploring the transcribed search space (section VII.B). An analysis of the relationship between manoeuvre types (i.e., defects and DSMs) is included to prove the robustness of the methodology in representing primary missions of interest (section VII.C).
- 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 VII.D). 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 transfer times between planets and number of revolutions between two consecutive encounters.
- Novel transfer options are explored in the context of a sample return mission towards comet 67P/Churyumov-Gerasimenko, the same target as Rosetta mission [64] (section VII.E).

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

- 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.
- Search spaces are substantially larger than those presented in similar problems in current literature.
- 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.

A. Multi-Objective Optimization of EVVEJS Sequence

The multi-objective optimization of Cassini-like EVVEJS sequence [2] is here assessed. The optimization is performed on the transcribed space as from section IV.A. In this case, the Δv computed as in Eq. 3 may lead to larger values. This is because they are manoeuvres applied immediately after departure from a fly-by, so are not representing DSMs in a real mission design context. A simple post-processing step as described in section VI is needed and again proves the robustness of the method in converging to benchmark solutions.

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

Table 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 scenario is described in Table 2 for a launch window in 1997. One should notice that transfer time bounds are chosen from available literature for benchmarking [‡] [21]. 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 transfers to the outermost planets since these are less sensitive to grid sizes due to their increased orbital periods [15].

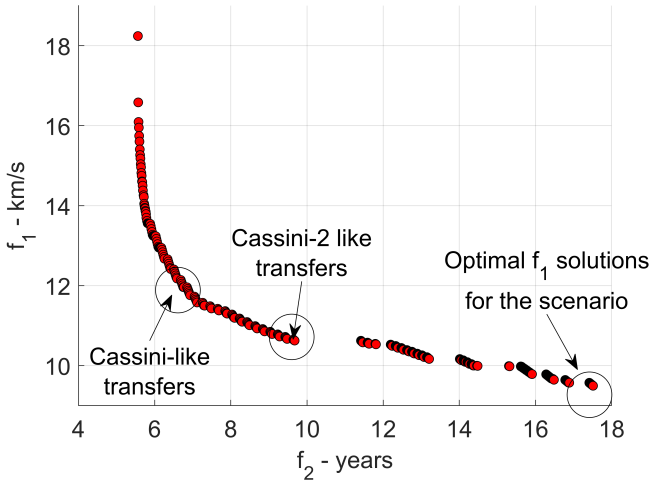


Fig. 9 Pareto front of f_1 and f_2 objectives as from MODP optimization for EVVEJS. Primary missions of interest for the scenario are highlighted.

The 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 section VII.B). One notices that a discretization using fractions (e.g. 1%) of the planets' orbital period would make the performance of the algorithm less dependent on the scale of the problem, and the range of geometry explored at each of the leg remains the same as in the fixed step sized case. In any case, the selected setup allows to obtain all the missions of interest for the given transfer scenario with a wide Pareto front and reduced effort (as in section VII.B), ranging from less than 6 years to almost 18 years of transfer time, as shown in Figure 9. The Pareto front is

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

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 duration of about 17.3 years.

Table 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. DEF stands for 'defect'.

Event	Cassini [2]	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
	−−	−−	−−
Venus fly-by	Apr. 21, 1998	May 22, 1998	Apr. 29, 1998
	DSM_2 = 0.466 km/s	DEF_2 = 1.97 km/s	DSM_2 = 0.431 km/s
Venus fly-by	Jun. 20, 1999	Jul. 01, 1999	Jun. 26, 1999
	−−	DEF_3 = 0.604 km/s	−−
Earth fly-by	Aug. 16, 1999	Aug. 20, 1999	Aug. 18, 1999
	−−	DEF_4 = 0.214 km/s	−−
Jupiter fly-by	Dec. 30, 2000	Jan. 01, 2001	Jan. 11, 2001
	−−	−−	−−
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

Details and trajectory representations can be found in Tables 3, 4 and 5, as well as in Figure 10, 11 and 12, 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 authors no reference solutions exist for the optimal sequences (Table 5 and Figure 12), again proving the efficiency of the pipeline in comprehensively solving the multi-objective optimization. The Pareto front in Figure 9 suggests that Cassini-2 solution is still constrained in terms of duration and significant improvements in terms of f_1 objective can be obtained. This is confirmed in Table 5, where the infinity velocity at Saturn is leveraged by a large DSM on the last leg at the price of an increased f_2 value (see also later section VII.C).

Table 4 Results for Cassini-2 like mission compared to solutions from defect model and refinement.

Event	Cassini-2 [§]	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
	DSM_1 = 0.480 km/s	−−	DSM_1 = 0.462 km/s
Venus fly-by	Apr. 29, 1998	May 20, 1998	Apr. 30, 1998
	DSM_2 = 0.398 km/s	DEF_2 = 1.83 km/s	DSM_2 = 0.398 km/s
Venus fly-by	Jun. 27, 1999	Jul. 02, 1999	Jun. 28, 1999
	−−	DEF_3 = 0.682 km/s	−−

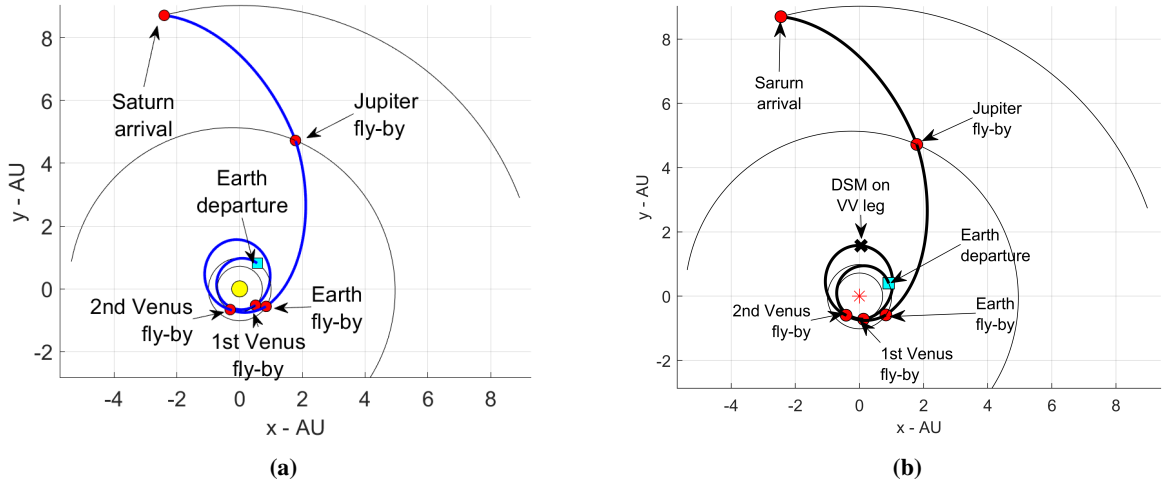
Continued on next page

Table 4 – *Continued from previous page*

Earth fly-by	Aug. 20, 1999	Aug. 21, 1999	Aug. 20, 1999
	−−	−−	−−
Jupiter fly-by	Mar. 31, 2001	Apr. 18, 2001	Apr. 01, 2001
	−−	−−	−−
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

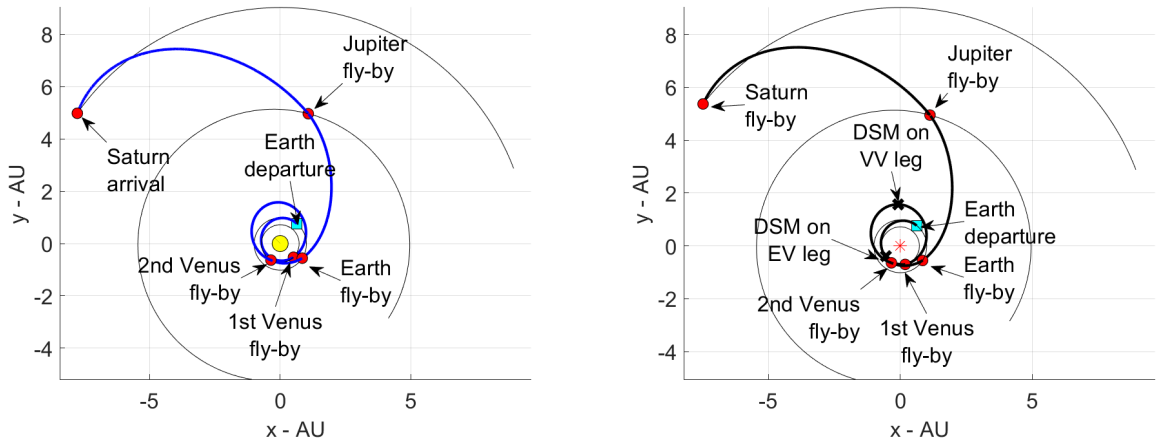
Table 5 Results for optimum solution in the given mission scenario compared to solutions from defect model and refinement.

Event	Defects solution	Refined solution
Earth departure	Nov. 13, 1997	Nov. 10, 1997
$v_{\infty,dep}$	3.63 km/s	3.59 km/s
	−−	DSM_1 = 0.694 km/s
Venus fly-by	May 10, 1998	May 01, 1998
	DEF_2 = 1.16 km/s	DSM_2 = 0.180 km/s
Venus fly-by	Jun. 02, 1999	Jun. 29, 1999
	DEF_3 = 0.0933 km/s	−−
Earth fly-by	Aug. 27, 1999	Aug. 26, 1999
	DEF_4 = 0.139 km/s	−−
Jupiter fly-by	Jun. 18, 2002	Jun. 12, 2002
	−−	DSM_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



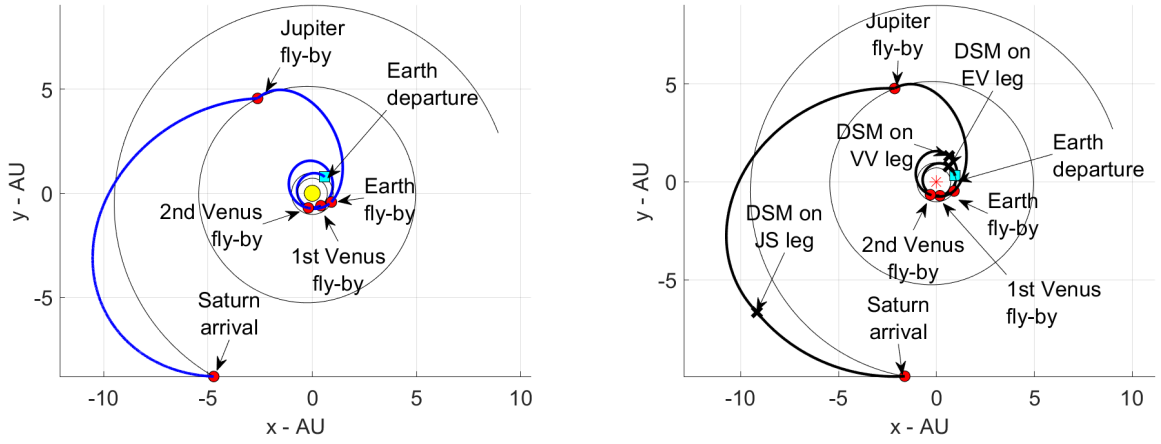
(a) **(b)**

Fig. 10 Cassini-like EVVEJS transfer with departure date in 1997 as resulting before (a) and after (b) the refinement process.



(a) **(b)**

Fig. 11 Cassini-2 EVVEJS transfer with departure date in 1997 as resulting before (a) and after (b) the refinement process.



(a) **(b)**

Fig. 12 Optimal solution for EVVEJS transfer scenario with departure date in 1997 as resulting before (a) and after (b) the refinement process.

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

In multi-objective optimization of the transcribed MGA trajectory design, a full evaluation (FE) of all possible combinations of departing dates and transfer times in an MGA sequence usually makes the number of possible routes to rise exponentially with the number of planets. This issue is mitigated via SODP and MODP approaches (see section V). 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 2. These are:

- The number of Lambert arcs evaluated (N_L)

- The number of defects computed (N_d)
- The number of routes stored 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 the maximum Δv defect at each encounter. Therefore a parametric study is 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 6 highlights the cases considered here. The N_r values represent the number of different routes for each sub-sequence terminating with the given leg. For example, for the leg VE, N_r is the number of routes of the sub-sequence EVVE.

Table 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

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 set characteristics are retained. This implies an upper limit on the DSM in-between two consecutive swing-bys, which is related by the leveraging ratio (i.e., ratio between Δv defect and precedent DSM). Table 7 represents the computational effort in terms of terms of N_L, N_d, $N_{r,FE}$. $N_{r,SODP}$ and $N_{r,MODP}$ with respect to the case considered.

Table 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	34,038	−−	2,249	2,249	2,249
VV	14,880	418,314	1,462	10,924	26,263
VE	11,718	271,932	4,764	66,249	1,190,503
EJ	64,160	1,910,364	1,772	28,170	4,864,556
JS	670,536	2,216,772	278,910	6,145,707	2,754,878,045
Totals	**795,332**	**4,817,382**	**278,910**	**6,145,707**	**2,754,878,045**
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	15,128	−−	995	995	995
VV	6,572	123,380	643	3,403	7,869

Continued on next page

Table 7 – *Continued from previous page*

VE	5,208	79,732	2,105	20,302	240,687
EJ	28,035	562,035	780	8,479	654,155
JS	292,734	650,520	121,213	1,754,661	246,966,060
Totals	**347,677**	**1,415,667**	**121,213**	**1,754,661**	**246,966,060**
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	5,550	−−	362	362	362
VV	2,475	27,150	225	798	1,642
VE	1,875	16,875	741	4,586	29,102
EJ	10,143	119,301	277	1,837	49,196
JS	103,707	138,777	42,454	340,078	11,146,629
Totals	**123,750**	**302,103**	**42,454**	**340,078**	**11,146,629**
Number of points in Pareto front				**205**	
Optimum for f_1				**9.566 km/s km/s**	

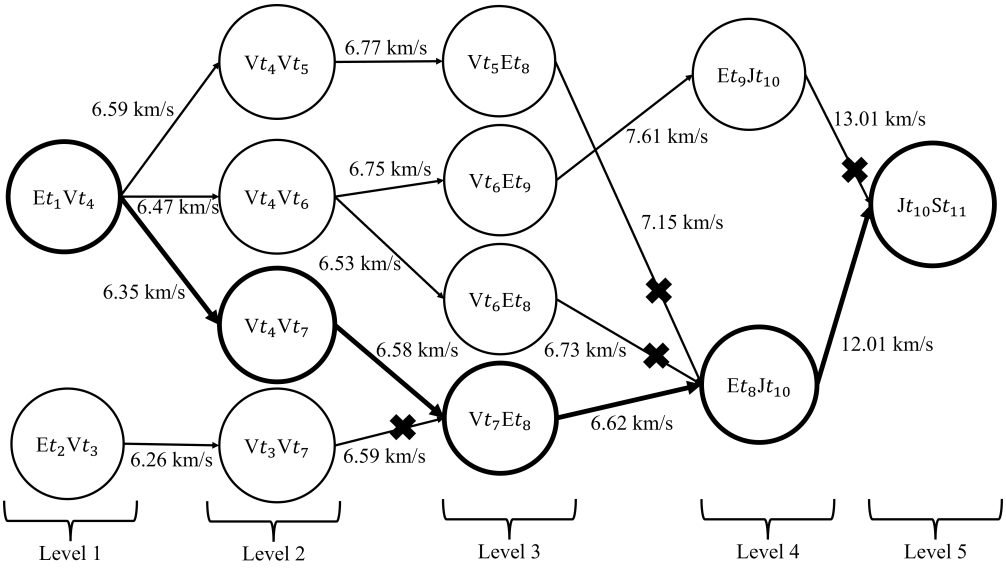


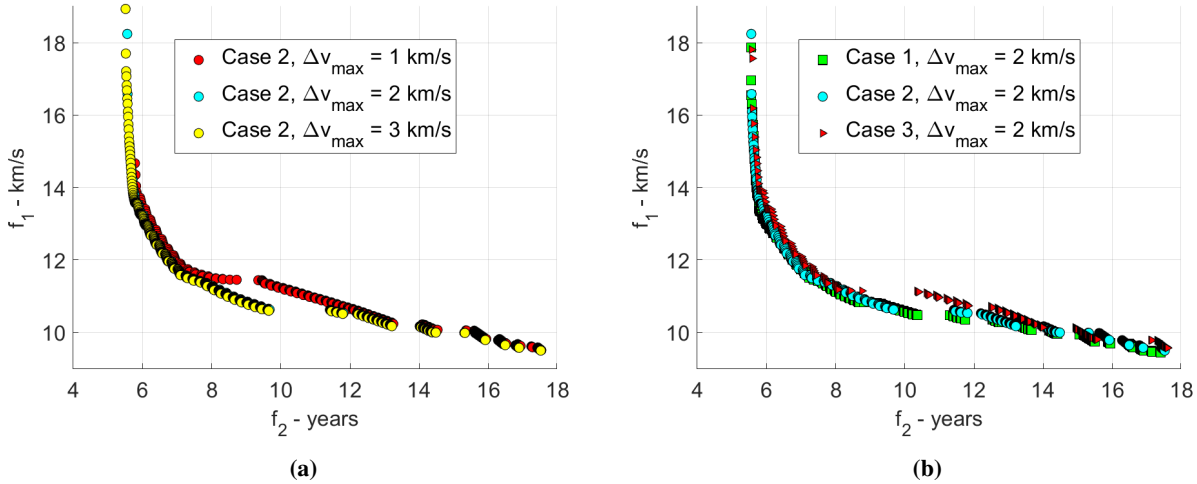
Fig. 13 Example of EVVEJS tree exploration. Crossed paths are pruned by SODP application. Accumulated Δv up to the given tree level are also shown. The bold path is f_1-optimal.

The number of Lambert problems solved N_L and the number of defects N_d decreases when coarser grids are considered. As can be seen, 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 13, 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 7 can also be used to infer approximate run times for the scenario considered, as the overall procedure is dominated by Lambert arcs and defects computations. Typically, around 1 microsecond is usually needed to compute a single Lambert arc [65–67] on standard laptop (i.e., 4 GHz single core), whilst for the defects computation times of only few nanoseconds are necessary. Therefore, no more than a 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. One notices that the number of evaluations for all of the legs depends upon the discretization scheme considered, and coarser grids would greatly reduce the effort.

One should also expect degradation of the Pareto front characteristics with larger step sizes and lower Δv_{max}, as shown Figure 14. It can be seen from Figure 14.a that increasing the defects limit to a value higher than 2 km/s does not produce significant variations of the Pareto front, whilst too aggressively pruning might result in imprecise Pareto set representation. The DSM leveraging ratio varies typically between 0.5 and 6.5 in most of the interplanetary missions considered here (see also section VII.C), 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, Δv_{max} = 2 km/s seems appropriate to truthfully represents Pareto front characteristics for the missions at hand. Figure 14.b highlights the impact of the step size with respect to the Pareto representation. A fine search (Case 2) is usually preferrable, if computationally acceptable, as coarser step sizes can degrade the solutions quality relatively quickly (Case 3). However, this choice is always a compromise between computational effort and solution quality, so that 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.



(a) **(b)**

Fig. 14 Pareto fronts for EVVEJS scenario varying with the Δv_{max} (a) and Case number (b).

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 VII.E) and are less consequential in terms of route numbers than other strategies. If another optimisation stage is envisaged, then a larger interval could also potentially be used (Case 3) with greater efficiency for the whole process.

C. Converting Defects into DSMs

A key feature of the pipeline described in this paper is the relationship between infinity velocity defects and DSMs. Therefore, assessing the relationship between these two manoeuvre types is important to the understanding of the whole pipeline. The analytical approximation described in section VI is employed here. This allows the derivation of the leveraging effect obtainable from a DSM, i.e., the dependency of an infinity velocity defect on a preceding DSM. As can be seen from both Table 3, 4 and 5, large defects occurring in the Venus-Venus leg of EVVEJS are replaced by DSMs, reducing from more than 1 km/s to approximately 400-600 m/s. The Δv occurring in all of 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 the optimal solution, while nominal Cassini experiences a 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 the high efficiency of the 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 15. The parameter represented is the ratio of the infinity velocity defect corrected to the magnitude of a DSM, versus the time 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 so 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 15.a and Figure 15.b), the leveraging ratio remains below 1 for all of the time elapsed in the segment, reaching its maximum value, i.e., 1, only at Saturn encounter. This suggests that a DSM is not 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 15.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 a large DSM between the last two planets as again proved by the refinement process in Table 5.

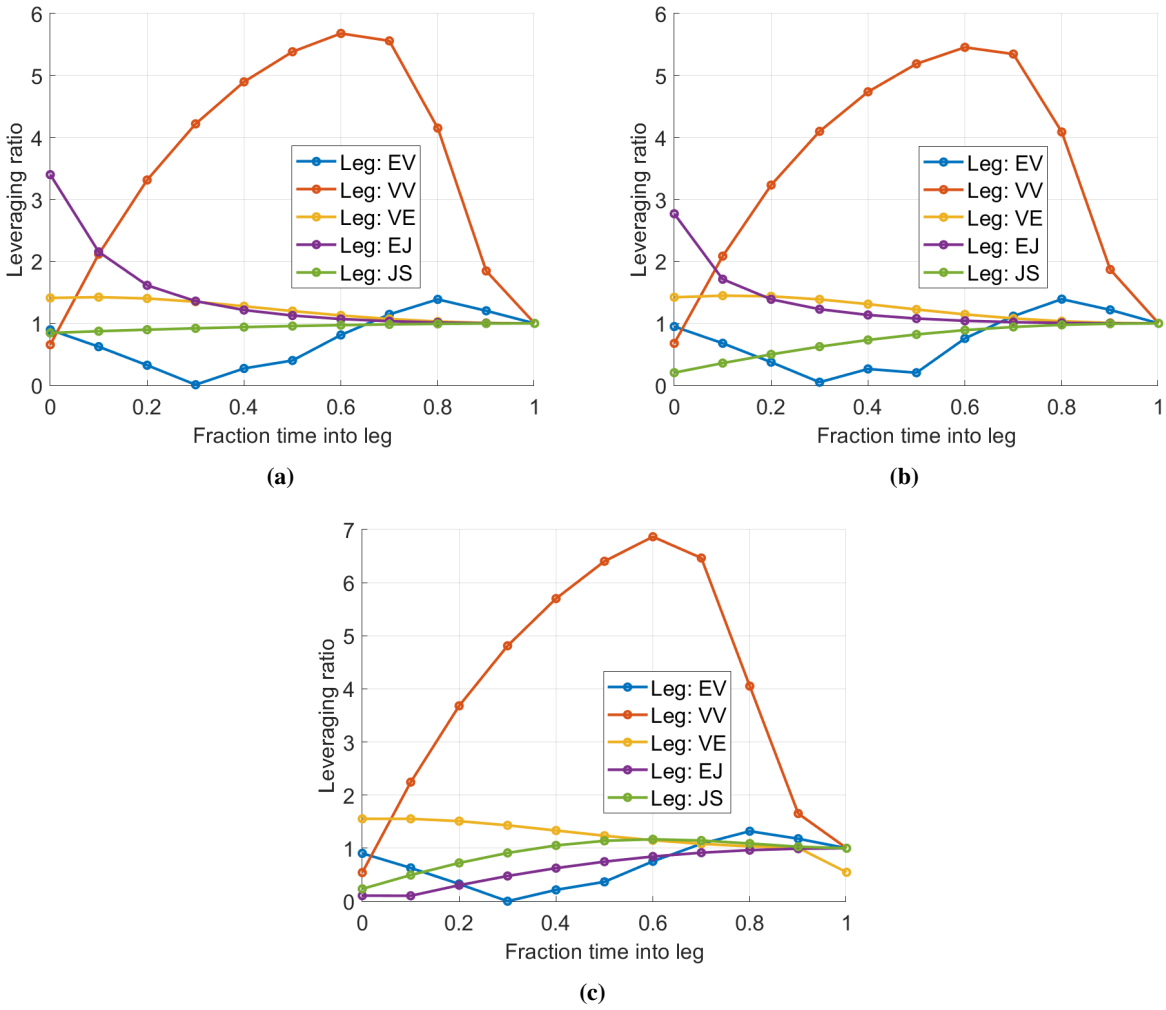


Fig. 15 Leveraging ratios per leg considered for Cassini (a), Cassini-2 (b) and optimal solution (c).

D. Multi-Objective Optimization of Earth-Saturn Missions

One then assumes that the sequence to reach Saturn is not known but needs to be selected as part of the optimization process. Table 8 highlights the set up for this scenario. One should notice that the time bounds are larger than those considered in Table 2, and multiple revolutions around the Sun (up until 2 full revolutions) are admissible for the innermost planets (i.e., if Venus, Earth, or Mars are involved in the transfer).

Table 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

Continued on next page

Table 8 – *Continued from previous page*

Number of revolutions about the Sun	$N_{rev} \leq 2$	
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, whose paths on Tisserand map are represented in Figure 16 and their Pareto fronts in Figure 17. Such Pareto fronts result from the specific mission scenario considered in Table 8, where less than one revolution is considered when either Jupiter or Saturn are present in a leg. One expects further reduction in f_1 in the region of the 24-25 years of transfer times if this hypothesis is removed.

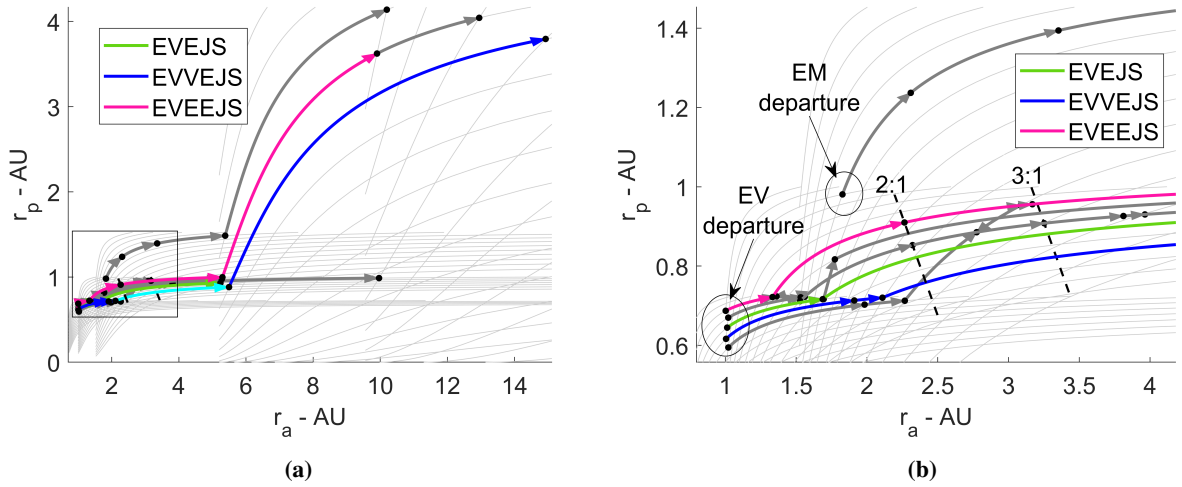


Fig. 16 Tisserand plots of Earth-Saturn options (in shaded grey) with interesting sequences highlighted (coloured lines). Squared region in plot (a) is zoomed in plot (b).

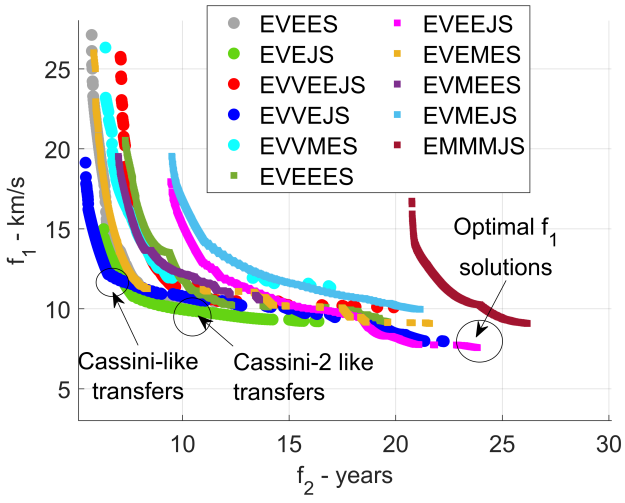


Fig. 17 Pareto front of Earth-Saturn options for the optimization scenario considered.

It can be noted that Cassini-like transfers following an EVVEJS sequence, in cyan in Figure 16, 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, in blue and orange in Figure 16, respectively, become competitive for longer mission durations, i.e., more than 10 years.

One can see that EVEJS seems to dominate EVVEJS for mission durations at about 10 years of transfer time. This mission option is characterized by a large defect in the EJ segment as reported in Table 9, as a single Earth fly-by usually is not sufficient 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 18 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 (i.e. to remove the velocity defect at the next Jupiter encounter). However, the defect at Jupiter is already relatively small, so the manoeuvre is likely not to be as effective as in the EVVEJS case, where high efficiency is seen in the VV leg, as demonstrated in the refinement step shown Table 9. Trajectories for EVEJS both before and after the refinement are shown in Figure 19, where one can see the manoeuvre at the very beginning of the EJ segment.

Table 9 Results for the ≈ 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
	−−	−−
Venus fly-by	Mar. 26, 1998	Mar. 24, 1998
	$DEF_2 = 0.425 km/s$	−−
Earth fly-by	Aug. 04, 1999	Aug. 02, 1999
	$DEF_3 = 1.63 km/s$	$DSM_2 = 1.44 km/s$
Jupiter fly-by	Apr. 09, 2001	May 12, 2001
	−−	−−
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, outperforming the missions identified in section VII.A, in terms of f_1 value, at the price of increased total transfer time f_2. The trajectory is depicted in Figure 20 and the corresponding values are shown in Table 10.

A general consideration is that the topology of the search space is correctly captured for any of the sequences identified (section VII.A), and the grid optimization alongside the dynamic programming approach already provides a very powerful tool to perform an accurate trade off analysis with very short computational effort (approximately 20 minutes on a 4 GHz laptop, see section VII.B). This is particularly useful in preliminary mission analysis (see also later section VII.E) when multiple mission options are often required with little work time allowed. Also there is very little a priori knowledge of the structure of the final trajectory. With this methodology, all the possible feasible trajectory

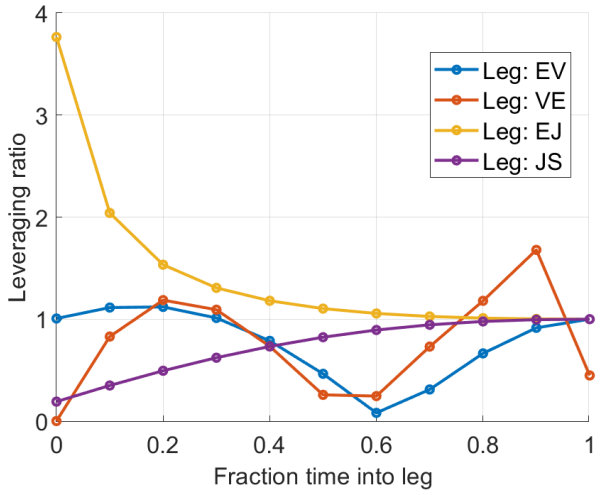


Fig. 18 Leveraging ratios per leg considered for EVEJS transfer for a departure date in 1997 and a transfer duration of about 10 years.

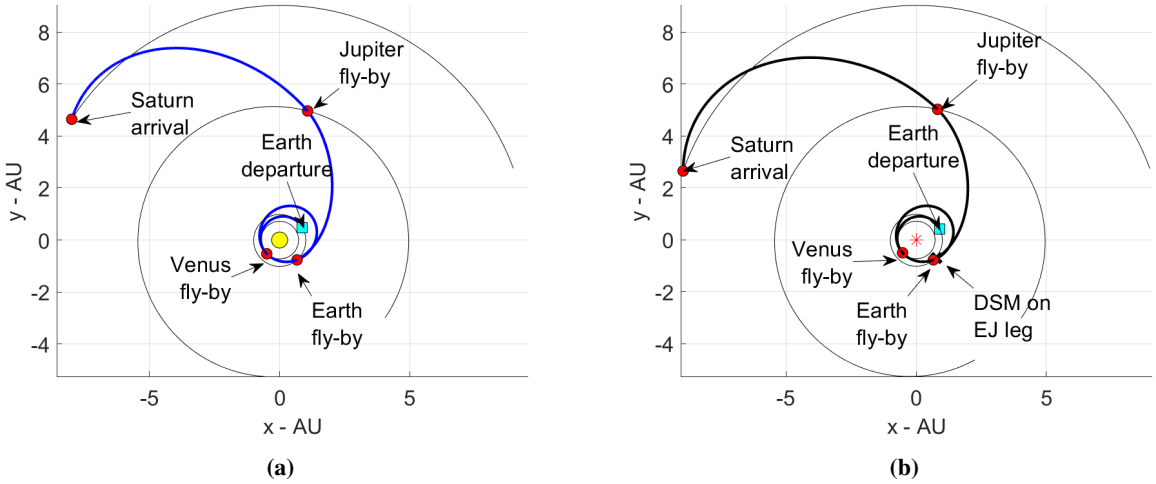


Fig. 19 EVEJS as from grid optimization (a) and refinement (b) for a departure date in 1997 and a transfer duration of about 10 years.

options are identified and can then efficiently initialise successive optimizations using higher-fidelity models.

Table 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
	−−	−−
Venus fly-by	Sep. 27, 1998	Oct. 10, 1998
	$DEF_2 = 0.609 km/s$	$DSM_2 = 0.525 km/s$
Earth fly-by	May 23, 2000	Jun. 16, 2000
	$DEF_3 = 0.0751 km/s$	$DSM_3 = 0.057 km/s$

Continued on next page

Table 10 – *Continued from previous page*

Earth fly-by	Feb. 14, 2007	Feb. 18, 2007
	−−	−−
Jupiter fly-by	May 08, 2011	May 11, 2011
	−−	−−
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

E. Comet Sample Return Missions

A mission scenario involving cometary objects amongst the Jupiter Family Comets (JFCs) is explored in this section. Such missions have been identified as future milestone in the context of the ESA Cosmic Vision 2050 [68]. The scenario considered here is reported in Table 11. The increased complexity of such scenario mainly results from the high number of object encounters, the extended launch window (between 2030-2040) and transfer times as well as the trajectory structure in terms of spacecraft revolutions about the Sun. The target comet is assumed to be known, and is 67P/Churyumov-Gerasimenko, i.e. the same target of ESA's Rosetta mission [64]. Compared to the scenarios in section VII.A and VII.D, the additional constraints considered for this optimization problem are: (1) 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, (2) the time t_w between cometary rendezvous and departure should be within 6 to 12 months to account for science phase operations.

Table 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	$N_{rev} \leq 2$	
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 $N_{rev} = 1$

The objective is to explore the whole 10-years launch window to find suitable trajectory options to go to the comet and return subject to the constraints. Thus, MODP-based approach is considered with the following objectives: (1) the

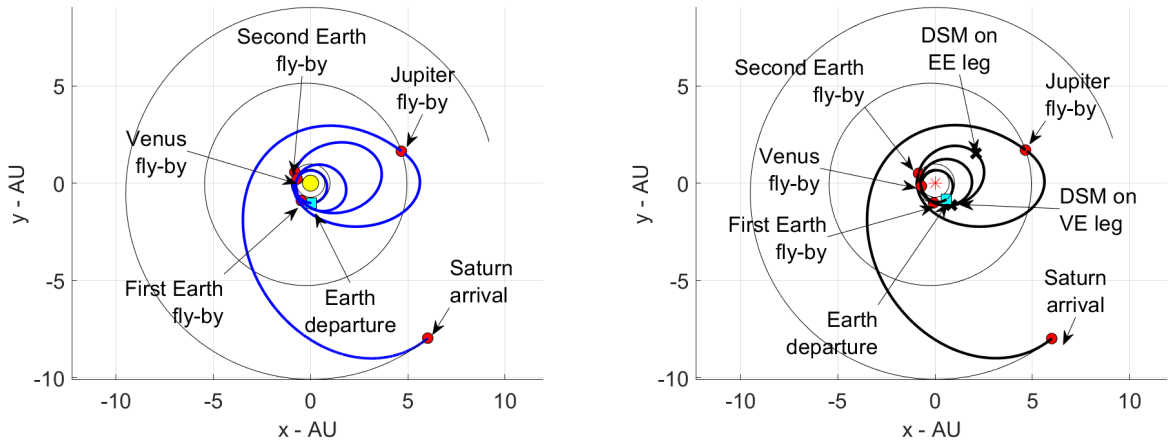


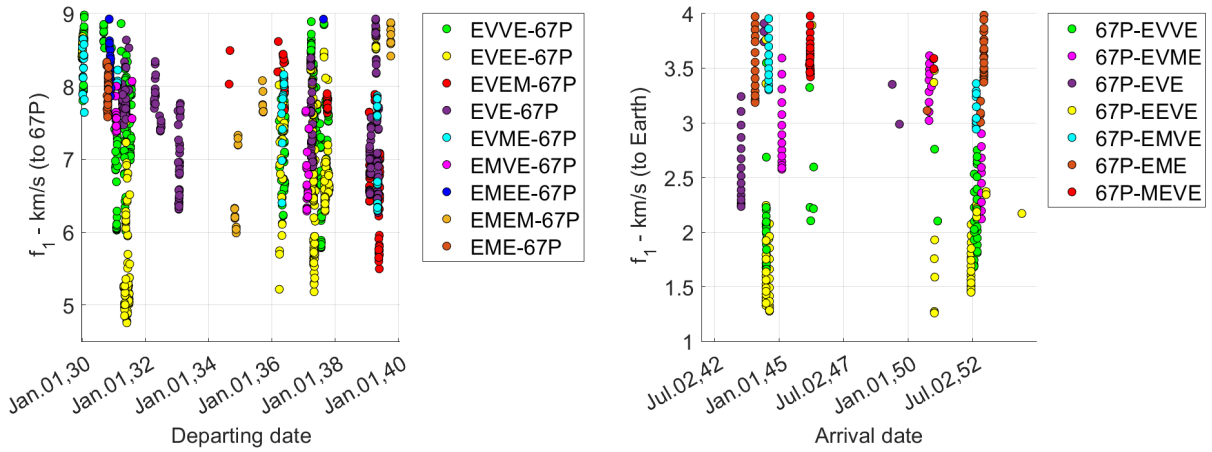
Fig. 20 EVEEJS as from grid optimization (a) and refinement (b) for a departure date in 1997 and a transfer duration of about 24 years.

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:

$$f_1 = \begin{cases} v_{\infty,dep} + \sum_{i=1}^{n_{int}-1} \Delta v_i + \Delta v_{arr} + \Delta v_{dep} + (v_{\infty,arr} - 4\ km/s) & \text{if } v_{\infty,arr} > 4 \text{ km/s} \\ v_{\infty,dep} + \sum_{i=1}^{n_{int}-1} \Delta v_i + \Delta v_{arr} + \Delta v_{dep} & \text{otherwise} \end{cases} \tag{12}$$

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 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 [69]). To ensure a suitable distribution 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. In other words, at each level of tree expansion, all those paths that arrive at a common node and have different departure dates are saved (i.e., in this case the path $(Et_2, Vt_3) - (Vt_3, Vt_7) - (Vt_7, Et_8)$ from Figure 13 would be saved alongside $(Et_1, Vt_4) - (Vt_4, Vt_7) - (Vt_7, Et_8)$).

The MODP exploration informed by Tisserand-based criterion automatically allows identification of a very high number of trajectories (109) involving up to 6 fly-bys with Solar System planets for a total of up to 10 objects in the overall sequence. The overall launch window exploration remains efficient (see section VII.B) and only takes approximately 1.5 hours (4 GHz laptop). Figure 21.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 21.b shows the cost of the return



(a) **(b)**

Fig. 21 Cost of transfers towards 67P (a) and back to the Earth (b) with respect to launch and arrival date, respectively.

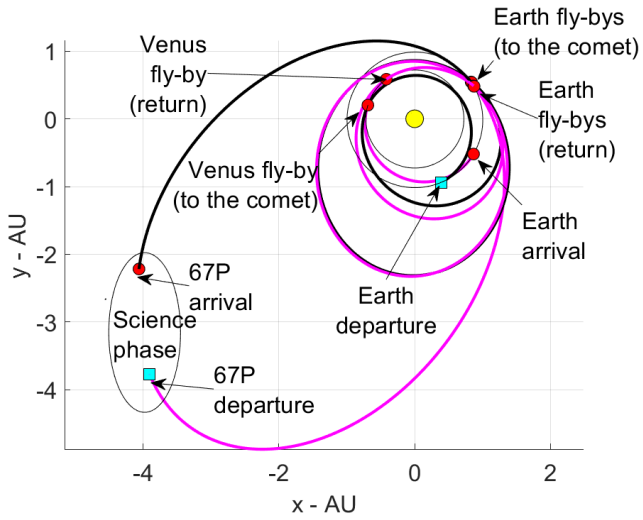


Fig. 22 Comet sample return trajectory to 67P involving an EVEE-67P trajectory to the comet (black path) and a 67P-EEVE transfer on the way back (magenta path).

phase with respect to the arrival date at the Earth. The two phases are here separated for the sake of representation, i.e., only those sequences that arrive to the comet and that can be completed with a return phase are shown, and the f_1 cost refers to the single phase under consideration. One notices that the reversed sequences like EVEE-67P and 67P-EEVE provide the cheapest transfers towards the comet and back to the Earth, respectively. This transfer is shown in Figure 22 and the corresponding events and values are reported in Table 12. No large manoeuvres are required either in the defect approximation, or in the refined solution, showing very good correspondence between defect solutions and refined ones.

Table 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
	$--$	$--$
Venus fly-by	Dec. 17, 2031	Dec. 17, 2031
	$--$	$--$
Earth fly-by	Oct. 26, 2032	Oct. 28, 2032
	$--$	$--$
Earth fly-by	Oct. 26, 2034	Oct. 28, 2034
	$DEF_4 = 0.743 km/s$	$--$
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
	$--$	$--$
Earth fly-by	Oct. 22, 2040	Oct. 22, 2040
	$--$	$--$
Earth fly-by	Oct. 22, 2042	Oct. 22, 2042
	$DEF_7 = 0.0874 km/s$	$--$
Venus fly-by	Mar. 13, 2044	Mar. 07, 2044
	$DEF_8 = 0.0262 km/s$	$--$
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

A general consideration is that the defect model will give the most accurate predictions for near-ballistic transfers with zero or low DSM requirement and shows greater divergence from fully optimized solutions where large DSMs are needed. The procedure has proven again to be efficient in exploring a mission scenario of complex configuration, demonstrating the ability to identify multiple mission options for practical preliminary analysis of future missions.

VIII. Conclusions

The paper presented a robust and efficient approach to the multi-objective optimization of complex MGA transfers. The pipeline employed uses processes that consider: (1) an energetic approach based on the Tisserand parameter in order to understand feasible MGA sequences; (2) 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; (3) refinement of selected solutions accounting for fly-by parameters and DSMs between two consecutive swing-bys, in the so-called MGA-DSM model. Specifically, robustness is ensured by the evaluation of the relationship between the different manoeuvre types. Dynamic programming is used to efficiently exploit the structure resulting from the MGA path planning step. In this way, multi-objective dynamic programming (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 to Jupiter and Saturn, similar to the JUICE and Cassini missions. The results obtained demonstrate the effectiveness in representing wide Pareto fronts with a complex configuration, for well-known trajectories. Indeed, in such cases the method allowed identification of the Δv-global optimum without any need for a priori knowledge of the solution (e.g., on the gravity-assist sequence, the departing date, transfer times and DSMs) in an efficient way. Wider search spaces are used compared to those in existing literature, as well as new transfer scenarios towards the aforementioned planets that populate very wide Pareto fronts. Numerical results show 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.

Appendix

The term $\frac{\partial|\vec{v}_{rel}|(t_f)}{\partial t_f}$ from Eq. (6) is shown to be zero by considering that:

$$(v_{rel}(t_f))^2 = \vec{v}_{rel} \cdot \vec{v}_{rel} = (\vec{v} - \vec{v}_p) \cdot (\vec{v} - \vec{v}_p) \tag{13}$$

where $\vec{v}$ and $\vec{v}_p$ are the spacecraft and planet velocities with respect to the central body, respectively, and $v_{rel} = |\vec{v}_{rel}|$. The partial derivative with respect to arrival time t_f is:

$$2v_{rel}(t_f)\frac{\partial v_{rel}(t_f)}{\partial t_f} = 2\vec{v}_{rel} \cdot \frac{\partial \vec{v}_{rel}}{\partial t_f} \tag{14}$$

Therefore, one has that:

$$\frac{\partial v_{rel}(t_f)}{\partial t_f} = \frac{\vec{v}_{rel} \cdot \frac{\partial \vec{v}_{rel}}{\partial t_f}}{v_{rel}(t_f)} \tag{15}$$

as well as:

$$\frac{\partial \vec{v}_{rel}}{\partial t_f} = \frac{\partial \vec{v}}{\partial t_f} - \frac{\partial \vec{v}_p}{\partial t_f} \tag{16}$$

The use of the partial derivative here signifies a partial where initial position is constant but final time is allowed to vary. In this case the above partials with respect to time are equivalent to the full-time derivatives of velocity. These time derivatives of planet and spacecraft velocity are evaluated at the point of rendezvous with the planet. Thus:

$$\frac{d\vec{v}}{dt_f} = \frac{d\vec{v}_p}{dt_f} = -\frac{\mu}{r_p^3}\vec{r}_p \tag{17}$$

i.e., both objects accelerate with the same gravitational acceleration as they are co-located. Therefore:

$$\frac{\partial \vec{v}}{\partial t_f} - \frac{\partial \vec{v}_p}{\partial t_f} = 0 \tag{18}$$

and so $\frac{\partial v_{rel}(t_f)}{\partial t_f} = 0$, using Eq. (15) and (16).

Acknowledgments

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

References

[1] D'Amario, L. A., Bright, L. E., and Wolf, A. A., "Galileo trajectory design," *Space Science Reviews*, Vol. 60, No. 1, 1992, pp. 23–78. https://doi.org/10.1007/BF00216849.

[2] Peralta, F., and Flanagan, S., "Cassini interplanetary trajectory design," *Control Engineering Practice*, Vol. 3, No. 11, 1995, pp. 1603–1610. https://doi.org/10.1016/0967-0661(95)00171-P.

[3] Yárnoz, D. G., Jehn, R., and Croon, M., "Interplanetary navigation along the low-thrust trajectory of BepiColombo," *Acta Astronautica*, Vol. 59, No. 1-5, 2006, pp. 284–293. https://doi.org/10.1016/j.actaastro.2006.02.028.

[4] Fox, N. J., Velli, M. C., Bale, S. D., Decker, R., Driesman, A., Howard, R. A., Kasper, J. C., Kinnison, J., Kusterer, M., and Lario, D., "The Solar Probe Plus Mission: Humanity's First Visit to Our Star," *Space Science Reviews*, Vol. 204, No. 1-4, 2016, pp. 7–48. https://doi.org/10.1007/s11214-015-0211-6.

[5] Sanchez Perez, J., Martens, W., and Varga, G., "Solar orbiter 2020 february mission profile," *Advances in the Astronautical Sciences*, Vol. 167, 2018, pp. 1395–1410.

[6] Grasset, O., Dougherty, M. K., Coustenis, A., Bunce, E. J., Erd, C., Titov, D., Blanc, M., Coates, A., Drossart, P., and Fletcher, L. N., "JUpiter ICy moons Explorer (JUICE): An ESA mission to orbit Ganymede and to characterise the Jupiter system," *Planetary and Space Science*, Vol. 78, 2013, pp. 1–21. https://doi.org/10.1016/j.pss.2012.12.002.

[7] Schlueter, M., "Nonlinear mixed integer based optimization technique for space applications," Ph.D. thesis, University of Birmingham, 2012.

[8] Schlueter, M., Erb, S. O., Gerdts, M., Kemble, S., and Rückmann, J.-J., "MIDACO on MINLP space applications," *Advances in Space Research*, Vol. 51, No. 7, 2013, pp. 1116–1131. https://doi.org/10.1016/j.asr.2012.11.006.

[9] Ross, I. M., and D'Souza, C. N., "Hybrid optimal control framework for mission planning," *Journal of Guidance, Control, and Dynamics*, Vol. 28, No. 4, 2005, pp. 686–697. https://doi.org/10.2514/1.8285.

[10] Longuski, J. M., and Williams, S. N., "Automated design of gravity-assist trajectories to Mars and the outer planets," *Celestial Mechanics and Dynamical Astronomy*, Vol. 52, No. 3, 1991, pp. 207–220. https://doi.org/10.1007/BF00048484.

[11] Sims, J. A., Staugler, A. J., and Longuski, J. M., "Trajectory options to Pluto via gravity assists from Venus, Mars, and Jupiter," *Journal of Spacecraft and Rockets*, Vol. 34, No. 3, 1997, pp. 347–353. https://doi.org/10.2514/2.3215.

[12] Petropoulos, A. E., Longuski, J. M., and Bonfiglio, E. P., "Trajectories to Jupiter via gravity assists from Venus, Earth, and Mars," *Journal of Spacecraft and Rockets*, Vol. 37, No. 6, 2000, pp. 776–783. https://doi.org/10.2514/2.3650.

[13] Izzo, D., Becerra, V. M., Myatt, D. R., Nasuto, S. J., and Bishop, J. M., "Search space pruning and global optimisation of multiple gravity assist spacecraft trajectories," *Journal of Global Optimization*, Vol. 38, No. 2, 2007, pp. 283–296. https://doi.org/10.1007/s10898-006-9106-0.

[14] Olympio, J. T., Marmorat, J.-P., and Izzo, D., "Global trajectory optimisation: can we prune the solution space when considering deep space maneuvers?" *ARIADNA study*, Vol. 6, 2007, p. 4101.

[15] Boutonnet, A., Martens, W., and Schoenmaekers, J., "SOURCE: A Matlab-oriented tool for interplanetary trajectory global optimization. Fundamentals," *Advances in the Astronautical Sciences*, Vol. 148, 2013, pp. 1449–1468.

[16] Abdelkhalik, O., and Mortari, D., "N-impulse orbit transfer using genetic algorithms," *Journal of Spacecraft and Rockets*, Vol. 44, No. 2, 2007, pp. 456–460. https://doi.org/10.2514/1.24701.

[17] Danoy, G., Dorronsoro, B., and Bouvry, P., "New State-of-the-art Results for Cassini2 Global Trajectory Optimization Problem," *Acta Futura*, Vol. 5, 2012, pp. 65–72. http://dx.doi.org/10.2420/AF05.2012.65.

[18] Olds, A. D., Kluever, C. A., and Cupples, M. L., "Interplanetary mission design using differential evolution," *Journal of Spacecraft and Rockets*, Vol. 44, No. 5, 2007, pp. 1060–1070. https://doi.org/10.2514/1.27242.

[19] Addis, B., Cassioli, A., Locatelli, M., and Schoen, F., "A global optimization method for the design of space trajectories," *Computational Optimization and Applications*, Vol. 48, 2011, pp. 635–652. https://doi.org/10.1007/s10589-009-9261-6.

[20] Choi, J. H., Lee, J., and Park, C., "Deep-space trajectory optimizations using differential evolution with self-learning," *Acta Astronautica*, Vol. 191, 2022, pp. 258–269. https://doi.org/10.1016/j.actaastro.2021.11.014.

[21] Vinkó, T., and Izzo, D., "Global optimisation heuristics and test problems for preliminary spacecraft trajectory design," *Advanced Concepts Team, ESATR ACT-TNT-MAD-GOHTPPSTD, Sept*, 2008.

[22] Pontani, M., and Conway, B. A., "Particle swarm optimization applied to impulsive orbital transfers," *Acta Astronautica*, Vol. 74, 2012, pp. 141–155. https://doi.org/10.1016/j.actaastro.2011.09.007.

[23] Vasile, M., Minisci, E., and Locatelli, M., "Analysis of some global optimization algorithms for space trajectory design," *Journal of Spacecraft and Rockets*, Vol. 47, No. 2, 2010, pp. 334–344. https://doi.org/10.2514/1.45742.

[24] Chilan, C. M., and Conway, B. A., "A space mission automaton using hybrid optimal control," *Advances in the Astronautical Sciences*, Vol. 127 PART 1, 2007, pp. 259–276.

[25] Wall, B. J., and Conway, B. A., "Genetic algorithms applied to the solution of hybrid optimal control problems in astrodynamics," *Journal of Global Optimization*, Vol. 44, No. 4, 2009, p. 493. https://doi.org/10.1007/s10898-008-9352-4.

[26] Englander, J., Conway, B., and Williams, T., "Automated interplanetary trajectory planning," *AIAA/AAS Astrodynamics Specialist Conference*, 2012, p. 4517. https://doi.org/10.2514/6.2012-4517.

[27] Englander, J. A., Conway, B. A., and Williams, T., "Automated mission planning via evolutionary algorithms," *Journal of Guidance, Control, and Dynamics*, Vol. 35, No. 6, 2012, pp. 1878–1887. https://doi.org/10.2514/1.54101.

[28] Wagner, S., and Wie, B., "Hybrid algorithm for multiple gravity-assist and impulsive delta-V maneuvers," *Journal of Guidance, Control, and Dynamics*, Vol. 38, No. 11, 2015, pp. 2096–2107. https://doi.org/10.2514/1.G000874.

[29] Ceriotti, M., and Vasile, M., "MGA trajectory planning with an ACO-inspired algorithm," *Acta Astronautica*, Vol. 67, No. 9-10, 2010, pp. 1202–1217. https://doi.org/10.1016/j.actaastro.2010.07.001.

[30] Gad, A., and Abdelkhalik, O., "Hidden genes genetic algorithm for multi-gravity-assist trajectories optimization," *Journal of Spacecraft and Rockets*, Vol. 48, No. 4, 2011, pp. 629–641. https://doi.org/10.2514/1.52642.

[31] Abdelkhalik, O., and Gad, A., "Dynamic-size multiple populations genetic algorithm for multigravity-assist trajectory optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 35, No. 2, 2012, pp. 520–529. https://doi.org/10.2514/1.54330.

[32] Vasile, M., Martin, J. M. R., Masi, L., Minisci, E., Epenoy, R., Martinot, V., and Baig, J. F., "Incremental planning of multi-gravity assist trajectories," *Acta Astronautica*, Vol. 115, 2015, pp. 407–421. https://doi.org/10.1016/j.actaastro.2015.05.033.

[33] Deb, K., Padhye, N., and Neema, G., "Interplanetary trajectory optimization with swing-bys using evolutionary multi-objective optimization," *International Symposium on Intelligence Computation and Applications*, Springer, 2007, pp. 26–35. https://doi.org/10.1007/978-3-540-74581-5_3.

[34] Vavrina, M. A., Englander, J. A., Phillips, S. M., and Hughes, K. M., "Global multi-objective trajectory optimization with parametric spreading," *Advances in the Astronautical Sciences*, Vol. 162, 2017, pp. 2151–2170.

[35] Englander, J. A., Vavrina, M. A., and Ghosh, A. R., "Multi-objective hybrid optimal control for multiple-flyby low-thrust mission design," *25th AAS/AIAA Space Flight Mechanics Meeting*, 2015, p. 227.

[36] Ellithy, A., Abdelkhalik, O., and Englander, J., "Multi-Objective Hidden Genes Genetic Algorithm for Multigravity-Assist Trajectory Optimization," *Journal of Guidance, Control, and Dynamics*, 2022, pp. 1–17. https://doi.org/10.2514/1.G006415.

[37] Acciarini, G., Izzo, D., and Mooij, E., "MHACO: a Multi-Objective Hypervolume-Based Ant Colony Optimizer for Space Trajectory Optimization," *2020 IEEE Congress on Evolutionary Computation (CEC)*, IEEE, 2020, pp. 1–8. https://doi.org/10.1109/CEC48606.2020.9185694.

[38] Zotes, F. A., and Penas, M. S., "Particle swarm optimisation of interplanetary trajectories from Earth to Jupiter and Saturn," *Engineering Applications of Artificial Intelligence*, Vol. 25, No. 1, 2012, pp. 189–199. https://doi.org/10.1016/j.engappai.2011.09.005.

[39] Vasile, M., and Zuiani, F., "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, 2011, pp. 1211–1227. https://doi.org/10.1177/0954410011410274.

[40] Bellman, R., "The theory of dynamic programming," *Bulletin of the American Mathematical Society*, Vol. 60, No. 6, 1954, pp. 503–515. https://doi.org/10.1090/S0002-9904-1954-09848-8.

[41] Strange, N. J., and Longuski, J. M., "Graphical method for gravity-assist trajectory design," *Journal of Spacecraft and Rockets*, Vol. 39, No. 1, 2002, pp. 9–16. https://doi.org/10.2514/2.3800.

[42] Vasile, M., and De Pascale, P., "Preliminary design of multiple gravity-assist trajectories," *Journal of Spacecraft and Rockets*, Vol. 43, No. 4, 2006, pp. 794–805. https://doi.org/10.2514/1.17413.

[43] Conway, B. A., *Spacecraft trajectory optimization*, Cambridge University Press, 2010. https://doi.org/10.1017/CBO9780511778025.

[44] Kemble, S., *Interplanetary mission analysis and design*, Springer Science & Business Media, 2006. https://doi.org/10.1007/3-540-37645-3.

[45] Olympio, J. T., and Izzo, D., "Designing optimal multi-gravity-assist trajectories with free number of impulses," *International Symposium on Space Flights Dynamics, ESA-ESTEC*, 2009.

[46] Peherstorfer, B., Willcox, K., and Gunzburger, M., "Survey of multifidelity methods in uncertainty propagation, inference, and optimization," *Siam Review*, Vol. 60, No. 3, 2018, pp. 550–591. https://doi.org/10.1137/16M1082469.

[47] de la Torre Sangrà, D., Fantino, E., Flores, R., Lozano, O. C., and Estelrich, C. G., "An automatic tree search algorithm for the Tisserand graph," *Alexandria Engineering Journal*, Vol. 60, No. 1, 2021, pp. 1027–1041. https://doi.org/10.1016/j.aej.2020.10.028.

[48] Heaton, A. F., Strange, N. J., Longuski, J. M., and Bonfiglio, E. P., "Automated design of the Europa Orbiter tour," *Journal of Spacecraft and Rockets*, Vol. 39, No. 1, 2002, pp. 17–22. https://doi.org/10.2514/2.3801.

[49] Kloster, K. W., Petropoulos, A. E., and Longuski, J. M., "Europa Orbiter tour design with Io gravity assists," *Acta Astronautica*, Vol. 68, No. 7-8, 2011, pp. 931–946. https://doi.org/10.1016/j.actaastro.2010.08.041.

[50] Campagnola, S., Buffington, B. B., and Petropoulos, A. E., "Jovian tour design for orbiter and lander missions to Europa," *Acta Astronautica*, Vol. 100, 2014, pp. 68–81. https://doi.org/10.1016/j.actaastro.2014.02.005.

[51] Colasurdo, G., Zavoli, A., Longo, A., Casalino, L., and Simeoni, F., "Tour of Jupiter Galilean moons: Winning solution of GTOC6," *Acta Astronautica*, Vol. 102, 2014, pp. 190–199. https://doi.org/10.1016/j.actaastro.2014.06.003.

[52] Campagnola, S., and Russell, R. P., "Endgame Problem Part 1: V_∞-Leveraging Technique and the Leveraging Graph," *Journal of Guidance, Control, and Dynamics*, Vol. 33, No. 2, 2010, pp. 463–475. https://doi.org/10.2514/1.44258.

[53] Campagnola, S., and Russell, R. P., "Endgame Problem Part 2: Multibody Technique and the Tisserand-Poincare Graph," *Journal of Guidance, Control, and Dynamics*, Vol. 33, No. 2, 2010, pp. 476–486. https://doi.org/10.2514/1.44290.

[54] Maiwald, V., "Applicability of Tisserand Criterion for Optimization of Gravity-assist Sequences for Low-thrust Missions," *Proceedings of the International Astronautical Congress, IAC*, 2015.

[55] Strange, N. J., Campagnola, S., and Russell, R. P., "Leveraging flybys of low mass moons to enable an enceladus orbiter," *Advances in the Astronautical Sciences*, Vol. 135, No. 3, 2009, pp. 2207–2225.

[56] AFSA, H., "AUTOMATE: Automatic Multi-Gravity Assist Trajectory Design with Tisserand Exploration," M.Sc. thesis, Cranfield University, 2021.

[57] Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C., *Introduction to algorithms*, MIT press, 2009.

[58] AFSA, H., Bellome, A., Sanchez Cuartielles, J. P., and Kemble, S., "Automatic multi-gravity assist trajectory design with modified Tisserand Graphs exploration," *Proceedings of the International Astronautical Congress, IAC*, 2022.

[59] Bellome, A., "Trajectory Design of Multi-Target Missions via Graph Transcription and Dynamic Programming," Ph.D. thesis, Cranfield University, 2022.

[60] Shapiro, S. C., *Encyclopedia of artificial intelligence second edition*, John Wiley and Sons, 1992.

[61] Ecale, E., Torelli, F., and Tanco, I., "JUICE interplanetary operations design: drivers and challenges," *15th International Conference on Space Operations*, 2018, p. 2493. http://doi.org/10.2514/6.2018-2493.

[62] Bellome, A., Sanchez Cuartielles, J. P., Del Ser, J., Kemble, S., and Felicetti, L., "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.

[63] Daellenbach, H. G., and De Kluyver, C. A., "Note on multiple objective dynamic programming," *Journal of the Operational Research Society*, 1980, pp. 591–594. https://doi.org/10.2307/2580846.

[64] Glassmeier, K.-H., Boehnhardt, H., Koschny, D., Kührt, E., and Richter, I., "The Rosetta Mission: Flying Towards the Origin of the Solar System," *Space Science Reviews*, Vol. 128, No. 1, 2007, pp. 1–21. https://doi.org/10.1007/s11214-006-9140-8.

[65] Gooding, R. H., "A procedure for the solution of Lambert's orbital boundary-value problem," *Celestial Mechanics and Dynamical Astronomy*, Vol. 48, No. 2, 1990, pp. 145–165. https://doi.org/10.1007/BF00049511.

[66] Izzo, D., "Revisiting Lambert's problem," *Celestial Mechanics and Dynamical Astronomy*, Vol. 121, No. 1, 2015, pp. 1–15. https://doi.org/10.1007/s10569-014-9587-y.

[67] Russell, R. P., "Complete Lambert Solver Including Second-Order Sensitivities," *Journal of Guidance, Control, and Dynamics*, Vol. 45, No. 2, 2022, pp. 196–212. https://doi.org/10.2514/1.G006089.

[68] Bockelée-Morvan, D., Filacchione, G., Altwegg, K., Bianchi, E., Bizzarro, M., Blum, J., Bonal, L., Capaccioni, F., Choukroun, M., and Codella, C., "AMBITION–comet nucleus cryogenic sample return," *Experimental Astronomy*, 2021, pp. 1–52. https://doi.org/10.1007/s10686-021-09770-4.

[69] Laipert, F., Nicholas, A., Olikara, Z., Woolley, R., and Lock, R., "Hybrid chemical-electric trajectories for a mars sample return orbiter," *Advances in the Astronautical Sciences*, Vol. 168, 2019, pp. 2847–2854.

2023-04-02

Multiobjective design of gravity-assist trajectories via graph transcription and dynamic programming

Bellome, Andrea

AIAA

Bellome A, Sánchez JP, Felicetti L, Kemble S. (2023) Multiobjective design of gravity-assist trajectories via graph transcription and dynamic programming, Journal of Spacecraft and Rockets, Available online 02 April 2023

https://doi.org/10.2514/1.A35472