Chapter 8
Global Optimization of Interplanetary Transfers with Deep Space Maneuvers Using Differential Algebra

Pierluigi Di Lizia, Roberto Armellin, Francesco Topputo, Franco Bernelli-Zazzera, and Martin Berz

Abstract In this chapter, differential algebra is used to globally optimize multi-gravity assist interplanetary trajectories with deep space maneuvers. A search space pruning procedure is adopted, and the trajectory design is decomposed into a sequence of sub-problems. As far as differential algebra is used, the objective function and the constraints are represented by Taylor series of the design variables over boxes in which the search space is divided. Thanks to the polynomial representation of the function and the constraints, a coarse grid can be used, and an efficient design space pruning is performed. The manipulation of the polynomials eases the subsequent local optimization process, so avoiding the use of stochastic optimizers. These aspects, along with the efficient management of the list of boxes, make differential algebra a powerful tool to design multi-gravity assist transfers including deep-space maneuvers.

Keywords Global optimization • Multi-gravity assist transfer • Deep-space maneuver • Search space pruning • Differential algebra

P. Di Lizia (✉) • R. Armellin • F. Topputo • F. Bernelli-Zazzera
Dipartimento di Ingegneria Aerospaziale, Politecnico di Milano, Via La Masa 34, 20156 Milano, Italy
e-mail: dilizia@aero.polimi.it; armellin@aero.polimi.it; topputo@aero.polimi.it; bernelli@aero.polimi.it

M. Berz
Department of Physics and Astronomy, Michigan State University, East Lansing MI 48824, USA
e-mail: berz@msu.edu

G. Fasano and J.D. Pintér (eds.), *Modeling and Optimization in Space Engineering*, Springer Optimization and Its Applications 73, DOI 10.1007/978-1-4614-4469-5_8,

List of Acronyms

DA	Differential algebra
DSM	Deep space maneuver
FP	Floating point
GASP	Gravity assist space pruning
MGA	Multi-gravity assist

8.1 Introduction

The preliminary design of impulsive interplanetary transfers is usually carried out in the frame of the patched-conics approximation. Within this context, different conic arcs are linked together to define the whole transfer trajectory. The patched-conics method allows the designer to define multiple gravity assist (MGA) transfers. MGA trajectories are usually made up of a sequence of planet-to-planet transfers in which the spacecraft exploits each planet encounter to achieve a velocity change. This method is well established in astrodynamics, and several past missions have used MGA trajectories to reach both inner and outer planets.

In the last two decades, mission designers have exploited the benefits of approaching complex MGA problems from a global optimization standpoint. Nowadays, the aim of the trajectory design is not only to find a solution, but also to find the best solution in terms of propellant consumption, while still achieving the mission goals. In the formalism of global optimization, this means that the problem consists in looking for the optimal solution in those regions of the search space that satisfy the problem constraints. Unfortunately, the MGA problems are characterized by an objective function with a large number of clustered minima, which are prevalently associated to the complex relative motion of the planets and to the nonlinearities governing the simple Kepler problem. This causes local optimization methods to converge to local minima. Hence, despite their efficiency, they should be avoided when looking for the global minimum of a MGA problem, at least in the first stage of the search process.

Extensive work has been devoted to address the global optimization of MGA transfers with impulsive maneuvers. This is mainly done by applying stochastic [1, 2, 3], branch and bound [4], meta-model-based [5], and combined [6] methods. Although some of them showed good performances, they tend to be computationally inefficient if not tailored on the MGA problem and on the structure of its search space.

The gravity assist space pruning (GASP) is a global optimization method that addresses this issue. GASP relies on a systematic evaluation of the objective and constraint functions on a grid of points distributed over the search space. The constraints are used to efficiently prune the search space [7]. Thanks to the particular class of interplanetary transfers solved by GASP, the planet-to-planet arcs making up the whole transfer are treated independently, and forward and backward constraining is applied. This way, the search space is preprocessed, and global optimization algorithms are employed in the reduced domain [8].

The class of MGA transfers formulated above does not cover all possible trajectories for a chemical-propelled spacecraft. An important option to take into account is the introduction of deep-space maneuvers (DSM). DSM are impulsive maneuvers, usually carried out between two planet encounters to improve the performances of a trajectory. When DSM are included into a MGA transfer, the resulting trajectory is usually referred to as MGA-DSM.

Unfortunately, pruning the solution space of MGA-DSM transfers is not trivial. First of all, the increased number of variables and the larger search space inhibits the use of a systematic approach to the pruning process. Moreover, local minima tend to proliferate, which makes difficult the detection of big prunable regions. Thus, it is necessary to rethink the whole pruning process implemented in GASP, and to reformulate it when DSM are included.

Differential algebra (DA) is proposed in this chapter as a valuable tool to address this task. Differential algebra serves the purpose of automatic differentiation, i.e., the accurate computation of the derivatives of functions in a computer environment. This goal is actually achieved by replacing the classical implementation of the real algebra with the proper implementation of a new algebra based on Taylor polynomials. Given a generic function f of v variables, the Taylor expansion of f up to any desired order k can be easily obtained from a computer algorithm that implements its evaluation.

The main idea behind the introduction of DA techniques into the pruning process is the substitution of the pointwise evaluation of the constraints, typical of GASP, with the computation of their Taylor expansions with respect to the design variables. The Taylor expansions are used to approximate the functions over boxes of the search space, and polynomial bounders are then exploited to estimate their ranges within each box. Consequently, the pointwise approach proposed in GASP can be substituted by a sampling process relying on box samples. This results in the possibility of enlarging the grid for the domain discretization, and reducing considerably the computational burden.

The chapter is organized as follows. A short description of the method underlying GASP is given in Sect. 8.2. Then, the implementation of a DA-based GASP algorithm is presented in Sect. 8.3. The introduction of DSM is addressed in Sect. 8.4. The performances of the resulting algorithm are assessed in Sect. 8.5, and some final remarks conclude the chapter.

8.2 Gravity Assist Space Pruning

An MGA transfer is modeled in GASP as a sequence of conic arcs, each patched to the subsequent one by a powered gravity assist maneuver. Consequently, a transfer involving n planets is a n-dimensional problem, as n epochs are needed to identify the position of the planets at each encounter. The main idea behind GASP is to split

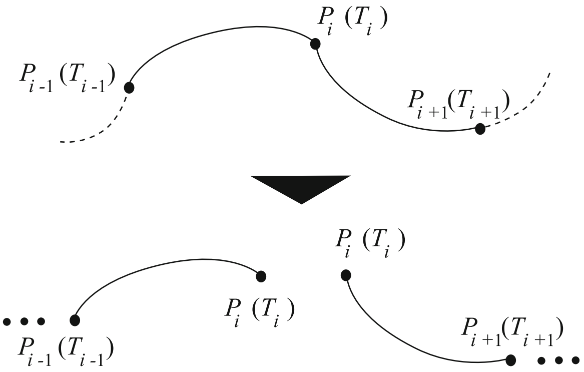


Fig. 8.1 Reduction of the MGA transfer to a cascade of two-dimensional subproblems

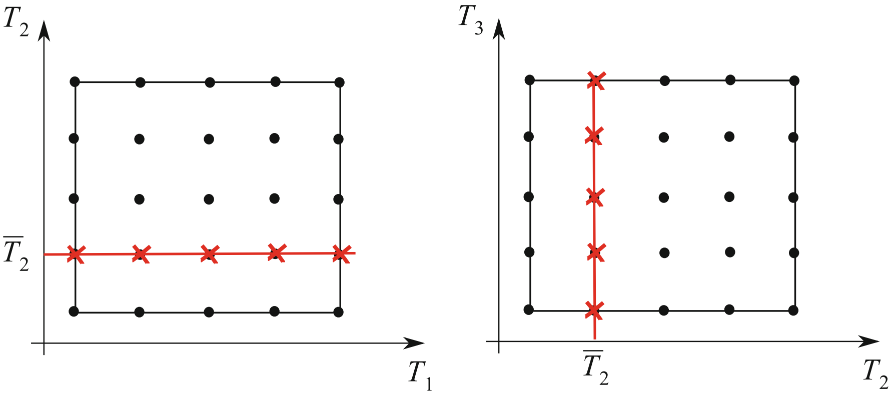


Fig. 8.2 Constraint propagation mechanism in GASP

the whole trajectory in its elementary arcs. With reference to Fig. 8.1, if P_i and T_i, $i = 1, \ldots, n$, are used to denote the planets and the epochs of the corresponding encounters, respectively, the arc connecting P_i to P_{i+1} can be treated as a two-dimensional subproblem with variables T_i and T_{i+1}.

For each subproblem, three constraints are imposed:

- Maximum ΔV at departure (first arc only) and arrival (last arc only).
- Maximum ΔV at gravity assist.
- Minimum pericenter radius at gravity assist.

These constraints can be profitably used to prune the search space. Consider, as an example, the first two arcs of an MGA transfer. These are characterized in the (T_1, T_2) and (T_2, T_3) spaces, respectively (see Fig. 8.2). A uniform grid of points is built to sample each of the search spaces. For each point in (T_1, T_2), the constraints of the P_1–P_2 transfer are evaluated. If any constraint is violated, the point is pruned away, together with all its subsequent combinations with the remaining epochs. In particular, if an entire row corresponding to $T_2 = \overline{T}_2$ yields unfeasible constraints in (T_1, T_2), the entire column corresponding to $T_2 = \overline{T}_2$ in (T_2, T_3) is pruned away. Similar statements hold for the subsequent arcs. This process is called forward constraint propagation. Analogously, a backward constraint propagation can be implemented. The final result is a reduced search space including only feasible regions, where optimization tools are run. The reduced dimension of the search space improves the performances of the optimization algorithms (the reader may refer to [7, 8] for details).

8.3 Gravity Assist Space Pruning with Differential Algebra

The use of differential algebra has been proposed in [9] to improve the performances of GASP. In the DA-based implementation of GASP, the search space is split into boxes, which are processed in place of grid points. More specifically, the point-wise evaluation of the constraint functions is substituted by the computation of their Taylor expansion over the sampling boxes. A polynomial bounder is then used to estimate the ranges of the functions within each box and to prune away unfeasible boxes. The formulation of the algorithm GASP into the DA framework is briefly described in this section. The reader may refer to [9] for additional details.

8.3.1 Notes on Differential Algebra

Differential algebra finds its origin in the attempt to solve analytical problems by an algebraic approach [10]. Historically, the treatment of functions in numerics has been based on the treatment of numbers, and the classical numerical algorithms are based on the mere evaluation of functions at specific points. DA techniques rely on the observation that it is possible to extract more information on a function than its mere values. The basic idea is to bring the treatment of functions and the operations on them to the computer environment in a similar way as the treatment of real numbers. Referring to Fig. 8.3, consider two real numbers a and b. Their transformation into the floating-point representation, $\overline{a}$ and $\overline{b}$, respectively, is performed to operate on them in a computer environment. Then, given any operation $*$ in the set of real numbers, an adjoint operation $\circledast$ is defined in the set of floating-point (FP) numbers so that the diagram in Fig. 8.3 commutes (The diagram commutes approximately in practice due to truncation errors.). Consequently, transforming the real numbers a and b into their FP representation and operating on them in the set of FP numbers returns the same result as carrying out the operation in the set of real numbers and then transforming the achieved result in its FP representation.

$$\begin{tikzcd}
a, b \in R \arrow[r, "\mathcal{T}"] \arrow[d, "*"'] & \bar{a}, \bar{b} \in FP \arrow[d, "\circledast"] \\
a * b \arrow[r, "\mathcal{T}"'] & \bar{a} \circledast \bar{b}
\end{tikzcd}
\qquad
\begin{tikzcd}
f, g \arrow[r, "\mathcal{P}"] \arrow[d, "*"'] & F, G \arrow[d, "\circledast"] \\
f * g \arrow[r, "\mathcal{P}"'] & F \circledast G
\end{tikzcd}$$

Fig. 8.3 Analogy between the floating-point representation of real numbers in a computer environment (*left figure*) and the introduction of the algebra of Taylor polynomials in the differential algebraic framework (*right figure*)

In a similar way, let us suppose two k-differentiable functions f and g in v variables are given. In the framework of differential algebra, the computer operates on them using their k-th order Taylor expansions, F and G, respectively. Therefore, the transformation of real numbers in their FP representation is now substituted by the extraction of the kth order Taylor expansions of f and g. For each operation in the space of k-differentiable functions, an adjoint operation in the space of Taylor polynomials is defined so that the corresponding diagram commutes, i.e., extracting the Taylor expansions of f and g and operating on them in the space of Taylor polynomials returns the same result as operating on f and g in the original space and then extracting the Taylor expansion of the resulting function.

The straightforward implementation of differential algebra in a computer allows to compute the Taylor coefficients of a function up to a specified order k, along with the function evaluation, with a fixed amount of effort. The Taylor coefficients of order k for sums and product of functions, as well as scalar products with reals, can be computed from those of summands and factors; therefore, the set of equivalence classes of functions can be endowed with well-defined operations, leading to the so-called truncated power series algebra [11, 12]. Similarly to the algorithms for floating-point arithmetic, the algorithms for functions follow, including methods to perform composition of functions, to invert them, to solve nonlinear systems explicitly, and to treat common elementary functions [10, 13]. In addition to these algebraic operations, the DA framework is endowed with differentiation and integration operators, therefore finalizing the definition of the DA structure. The differential algebra sketched in this section is implemented in the software COSY-Infinity [14].

For the sake of a more comprehensive illustration of the DA basics, the next section introduces the simplest nontrivial differential algebra for the first-order expansion of univariate functions. The reader can refer to [10] for its extension to the arbitrary order expansion of multivariate functions.

8.3.1.1 The Minimal Differential Algebra

Consider all ordered pairs (q_0, q_1), with q_0 and q_1 real numbers. Define addition, scalar multiplication, and vector multiplication as follows:

$$\begin{aligned}(q_0, q_1) + (r_0, r_1) &= (q_0 + r_0, q_1 + r_1),\\ t \cdot (q_0, q_1) &= (t \cdot q_0, t \cdot q_1),\\ (q_0, q_1) \cdot (r_0, r_1) &= (q_0 \cdot r_0, q_0 \cdot r_1 + q_1 \cdot r_0).\end{aligned} \tag{8.1}$$

The ordered pairs with the above arithmetic are called ${}_1D_1$. The multiplication of vectors is seen to have (1, 0) as the unity element. The multiplication is commutative, associative, and distributive with respect to addition. Together, the three operations defined in Eq. (8.1) form an algebra. Furthermore, they form an extension of real numbers, as $(r, 0) + (s, 0) = (r + s, 0)$ and (r, 0) ·(s, 0) = (r ·s, 0), so that the reals are included.

The multiplicative inverse of the pair (q_0, q_1) in ${}_1D_1$ is

$$(q_0, q_1)^{-1} = \left(\frac{1}{-q_0}, -\frac{q_1}{q_0^2}\right), \tag{8.2}$$

which is defined for any $q_0 \neq 0$.

One important property of this algebra is that it has an order compatible with its algebraic operations. Given two elements (q_0, q_1) and (r_0, r_1) in ${}_1D_1$, it is defined

$$\begin{aligned}&(q_0, q_1) < (r_0, r_1) \quad \textit{if} \quad q_0 < r_0 \quad \textit{or} \quad (q_0 = r_0 \text{ and } q_1 < r_1),\\ &(q_0, q_1) > (r_0, r_1) \quad \textit{if} \quad (r_0, r_1) \ < (q_0, q_1),\\ &(q_0, q_1) = (r_0, r_1) \quad \textit{if} \quad q_0 = r_0 \ \text{ and } q_1 = r_1.\end{aligned} \tag{8.3}$$

As for any two elements (q_0, q_1) and (r_0, r_1) only one of the three relation holds, ${}_1D_1$ is said totally ordered. The order is compatible with the addition and multiplication; for all (q_0, q_1), (r_0, r_1), (s_0, s_1) ∈ ${}_1D_1$, it follows (q_0, q_1) < (r_0, r_1) ⇒ $(q_0, q_1) + (s_0, s_1) < (r_0, r_1) + (s_0, s_1)$, and ($s_0$, s_1) > (0, 0) = 0 ⇒ (q_0, q_1) ·(s_0, s_1) < (r_0, r_1) ·(s_0, s_1).

The number d = (0, 1) has the interesting property of being positive but smaller than any positive real number; indeed (0, 0) < (0, 1) < (r, 0) = r. For this reason d is called an infinitesimal or a differential. In fact, d is so small that its square vanishes. Since for any (q_0, q_1) $\in {}_1D_1$

$$(q_0, q_1) = (q_0, 0) + (0, q_1) = q_0 + d \cdot q_1, \tag{8.4}$$

the first component is called the real part and the second component the differential part.

The algebra in ${}_1D_1$ becomes a differential algebra by introducing a map ∂ from ${}_1D_1$ to itself, and proving that the map is a derivation. Define $\partial : {}_1D_1 \to {}_1D_1$ by

$$\partial(q_0, q_1) = (0, q_1). \tag{8.5}$$

Note that

$$\begin{aligned}\partial\{(q_0, q_1) + (r_0, r_1)\} &= \partial(q_0 + r_0, q_1 + r_1) = (0, q_1 + r_1)\\ &= (0, q_1) + (0, r_1) = \partial(q_0, q_1) + \partial(r_0, r_1)\end{aligned} \tag{8.6}$$

and

$$\begin{aligned}\partial\{(q_0, q_1) \cdot (r_0, r_1)\} &= \partial(q_0 \cdot r_0, q_0 \cdot r_1 + r_0 \cdot q_1) = (0, q_0 \cdot r_1 + r_0 \cdot q_1)\\ &= (0, q_1) \cdot (r_0, r_1) + (0, r_1) \cdot (q_0, q_1)\\ &= \partial\{(q_0, q_1)\} \cdot (r_0, r_1) + (q_0, q_1) \cdot \partial\{(r_0, r_1).\}\end{aligned} \tag{8.7}$$

This holds for all $(q_0, q_1), (r_0, r_1) \in {}_1D_1$. Therefore, ∂ is a derivation and $({}_1D_1, \partial)$ is a differential algebra.

The most important aspect of ${}_1D_1$ is that it allows the automatic computation of derivatives. Assume to have two functions f and g and to put their values and their derivatives at the origin in the form $(f(0), f'(0))$ and $(g(0), g'(0))$ as two vectors in ${}_1D_1$. If the derivative of the product $f \cdot g$ is of interest, it has just to be looked at the second component of the product $(f(0), f'(0)) \cdot (g(0), g'(0))$, whereas the first component gives the value of the product of the functions. Therefore, if two vectors contain the values and the derivatives of two functions, their product contains the values and the derivatives of the product function. Defining the operator [ ] from the space of differential functions to ${}_1D_1$ via

$$[f] = (f(0), f\prime(0)), \tag{8.8}$$

it holds

$$\begin{aligned}[f + g] &= [f] + [g],\\ [f \cdot g] &= [f] \cdot [g]\end{aligned} \tag{8.9}$$

and

$$[1/g] = [1]/[g] = 1/[g] \tag{8.10}$$

by using (8.2). This observation can be used to compute derivatives of many kinds of functions algebraically by merely applying arithmetic rules on ${}_1D_1$, beginning from the value and the derivative of the identity function $[x] = (x, 1) = x + \delta x$. Consider the example

$$f(x) = \frac{1}{x + (1/x)} \tag{8.11}$$

and its derivative

$$f'(x) = \frac{(1/x^2) - 1}{(x + (1/x))^2}. \tag{8.12}$$

The function value and its derivative at the point $x = 3$ are

$$f(3) = \frac{3}{10}, \quad f'(3) = -\frac{2}{25}. \tag{8.13}$$

Evaluating the function (8.11) in the DA framework at $(3,1) = 3 + \delta x$ yields

$$\begin{aligned} f((3,1)) &= \frac{1}{(3,1) + 1/(3,1)} = \frac{1}{(3,1) + (1/3, -1/9)} \\ &= \frac{1}{(10/3, 8/9)} = \left(\frac{3}{10}, -\frac{8}{9} \Big/ \frac{100}{9}\right) = \left(\frac{3}{10}, -\frac{2}{25}\right). \end{aligned} \tag{8.14}$$

Thus, the real part of the result is the value of the function at $x = 3$, whereas the differential part is the value of the derivative of the function at $x = 3$. This is simply justified by applying the relations (8.9) and (8.10).

$$\begin{aligned} [f(x)] &= \left[\frac{1}{x + 1/x}\right] = \frac{1}{[x + 1/x]} \\ &= \frac{1}{[x] + [1/x]} = \frac{1}{[x] + 1/[x]} \\ &= f([x]). \end{aligned} \tag{8.15}$$

The method can be generalized to treat common intrinsic functions.

8.3.2 Representation of Objective and Constraint Functions

The evaluation of the objective and constraint functions in MGA transfers involves solving implicit equations, which become parametric when their Taylor expansion in the design variables is of interest. Three implicit equations have to be solved in the model adopted. Two of them already appear in simple planet-to-planet transfers. These are illustrated with a practical example in the following. Let us consider the transfer from planet P_1 to planet P_2 sketched in Fig. 8.4.

The objective function for this problem is the overall ΔV; this can be evaluated by using two design variables. A common choice is selecting the departure epoch

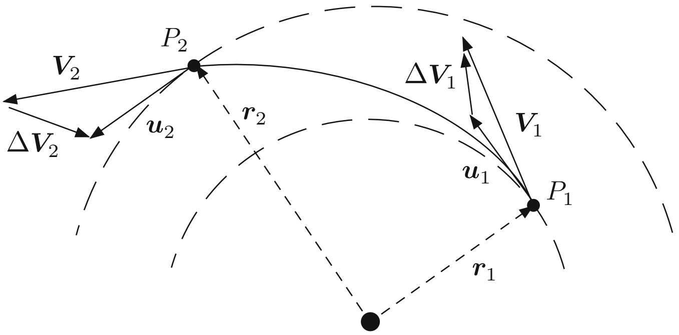


Fig. 8.4 A two-impulse planet-to-planet transfer

from P_1, T_1, and the time of flight, t_{12}. The arrival epoch at P_2 is $T_2 = T_1 + t_{12}$, and the position and velocity of P_1 and P_2 at both ends of the transfer ($\mathbf{r}_1$, $\mathbf{v}_1$ and $\mathbf{r}_2$, $\mathbf{v}_2$, respectively) are obtained through their planetary ephemerides. Given $\mathbf{r}_1$, $\mathbf{r}_2$, and t_{12}, the corresponding Lambert's problem is solved to compute the heliocentric initial and final velocities, $\mathbf{V}_1$ and $\mathbf{V}_2$, respectively. The two velocity impulses required to accomplish the transfer are $\Delta V_i = \mathbf{V}_i - \mathbf{v}_i$, $i = 1, 2$.

Problem Statement. Let $\mathbf{x} = \{T_1, t_{12}\}$, optimal two-impulse transfers from P_1 to P_2 are found by solving

$$\min_{\mathbf{x}} \Delta V(\mathbf{x}) \quad \textit{subject to} \quad \begin{aligned} \Delta V_1(\mathbf{x}) &\le \Delta V_1^{max} \\ \Delta V_2(\mathbf{x}) &\le \Delta V_2^{max}, \end{aligned} \tag{8.16}$$

where $\Delta V = \Delta V_1 + \Delta V_2 = \|\Delta V_1\| + \|\Delta V_2\|$ and $\mathit{\Delta} V_1{}^{\max}$ and $\mathit{\Delta} V_2{}^{\max}$ are maximum allowed values for $\mathit{\Delta} V_1$ and $\mathit{\Delta} V_2$, respectively.

The evaluation of planetary ephemerides is required to compute $\mathbf{r}_i$ and $\mathbf{v}_i$, for $i = 1, 2$. An analytical ephemeris model is used, which is based on interpolating the planetary orbital elements delivered by JPL's Horizons system [15] with cubic splines. The analytical model supplies the eccentricity of the planet orbit, e, and the mean anomaly of the planet, M, at the evaluation epoch. Then, the Kepler equation

$$f(E) = E - e\sin E - M = 0\ , \tag{8.17}$$

must be solved for the eccentric anomaly, E, which is necessary to evaluate the planet position and velocity.

The second implicit equation appears in the solution of the Lambert problem for $\mathbf{V}_1$ and $\mathbf{V}_2$. In Lambert's problem, the initial position, final position, and the time of flight between the two positions are given. Solving Lambert's problem defines the Keplerian orbit that connects the two position vectors in the given time, allowing the calculation of the velocities at the initial and final positions. Lambert's theorem states that the time of flight $\Delta t = t_2 - t_1$ depends only on the semi-major axis a, the sum of the two radii $r_1 + r_2$, and the distance between the initial and final positions, i.e., the chord length $c = ||r_2 - r_1||$ [16]. The time required for the transfer can be written as

$$\Delta t = \sqrt{\frac{a^3}{\mu}}(2k\pi + (E_2 - e\sin E_2) - (E_1 - e\sin E_1)), \tag{8.18}$$

where E_1 and E_2 are the eccentric anomalies of the initial and final positions respectively, measured on the connecting arc. The problem, now is to find the correct values of a, E_1, E_2, and e that give the desired time of flight. As Lambert stated, however, the transfer time depends only on the three quantities mentioned earlier. The two radii and the chord length are already known from the problem definition. The semi-major axis is the only unknown parameter. Thus, it is possible to write the transfer time as a function of the semi-major axis only, or some other parameters such as p or ΔE. In our approach, based on Battin's algorithm [16], the nonlinear equation to be solved is

$$A(x) - \Delta t = 0, \tag{8.19}$$

in which

$$A(x) = g(x)^{3/2}(\alpha(x) - \sin\alpha(x) - \beta(x) + \sin\beta(x)). \tag{8.20}$$

The functions $\alpha(x)$ and $\beta(x)$ are related to x via the relations

$$\sin^2\frac{1}{2}\alpha(x) = \frac{s}{2g(x)} \qquad \sin^2\frac{1}{2}\beta(x) = \frac{s-c}{2g(x)}, \tag{8.21}$$

with

$$g(x) = \frac{s}{2(1-x^2)}, \tag{8.22}$$

and the semi-perimeter

$$s = (r_1 + r_2 + c)/2. \tag{8.23}$$

Note that the relation between a and x is simply given by

$$a = \frac{s}{2(1-x^2)}. \tag{8.24}$$

Once Eq. (8.19) is solved, the initial and final heliocentric velocities of the spacecraft are computed via algebraic and transcendental functions.

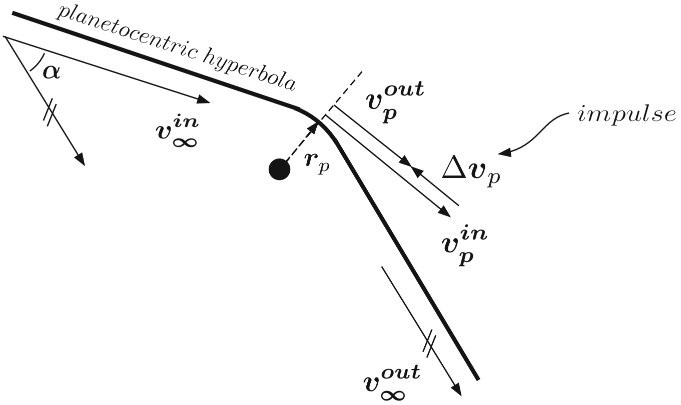


Fig. 8.5 Powered gravity assist

The third implicit equation occurs when transfers with powered gravity assists are considered. In a powered gravity assist, the spacecraft provides a tangential impulse at the pericenter of the incoming hyperbola. Therefore, the planeto-centric trajectory is made up of two arcs of hyperbola patched together (see Fig. 8.5). The angle α, usually referred to as bending angle, between the incoming and the outgoing asymptotic velocities, $v_\infty{}^{\mathrm{in}}$ and $v_\infty{}^{\mathrm{out}}$, respectively, is related to the pericenter radius via [8]

$$f(r_p) = \arcsin\frac{a^-}{a^- + r_p} + \arcsin\frac{a^+}{a^+ + r_p} - \alpha = 0\ , \tag{8.25}$$

where $a^- = 1/(\mathbf{v}_\infty^{in} \cdot \mathbf{v}_\infty^{in})$ and $a^+ = 1/(\mathbf{v}_\infty^{out} \cdot \mathbf{v}_\infty^{out})$. The angle α can be easily computed from the two heliocentric arcs connected at the gravity assist. The solution of the implicit equation (8.25) delivers the pericenter radius of the planetocentric trajectory. The planetocentric velocities $\boldsymbol{v}_p{}^{\mathrm{in}}$ and $\boldsymbol{v}_p{}^{\mathrm{out}}$ at the pericenter, corresponding to the incoming and outgoing hyperbolic arcs, respectively, are computed using $\boldsymbol{r}_p$, $\boldsymbol{v}_\infty{}^{\mathrm{in}}$, and $\boldsymbol{v}_\infty{}^{\mathrm{out}}$. Thus, the magnitude of the impulsive maneuver at the pericenter is simply $\Delta v_p = \|\mathbf{v}_p^{out} - \mathbf{v}_p^{in}\|$.

A classical numerical method for the solution of implicit equations can be used to solve Eqs. (8.17)–(8.25) for a point-wise evaluation of the objective and constraint functions. This is not true when the Taylor expansion of the objective and constraint functions is of interest, as the implicit equations become parametric in the design variables. This is briefly illustrated for Eq. (8.17) in the following. Similar arguments hold for Eqs. (8.19) and (8.25).

Let us consider the evaluation of planetary ephemerides. In the DA framework, we are interested in the Taylor expansion of planet's position and velocity with respect to the evaluation epoch. Thus, Kepler's equation (8.17) is not solved for real values of the eccentric anomaly, but rather for its Taylor expansion with respect to the epoch. More specifically, the epoch is initialized as a DA variable, $[T] = T^0$

$+\delta T$, where δT is the displacement of the epoch from the reference value T^0. Then, the simple evaluation of the analytical ephemeris model in the DA framework delivers the Taylor expansion of the eccentricity e and the mean anomaly M with respect to the epoch,

$$\begin{aligned} [e] &= \mathcal{M}_e(\delta T), \\ [M] &= \mathcal{M}_M(\delta T), \end{aligned} \tag{8.26}$$

where $\mathcal{M}_e$ and $\mathcal{M}_M$ denote the resulting Taylor polynomials for e and M. Thus, the explicit dependence of e and M on δT appears in Kepler's equation, which now reads

$$f(E, \delta T) = E - [e] \sin E - [M] = E - \mathcal{M}_e(\delta T) \sin E - \mathcal{M}_M(\delta T) = 0. \tag{8.27}$$

The parametric implicit equation (8.27) must be solved for the Taylor expansion of E with respect to the parameter δT, $[E] = \mathcal{M}_E(\delta T)$. Dedicated techniques have been developed in past works to address the previous task [9]. Once $\mathcal{M}_E(\delta T)$ is available, the Taylor expansions of the planet position and velocity are readily obtained by carrying out the remaining algebra in the DA framework.

8.3.3 Implementation of GASP-DA

The use of differential algebra is now introduced in GASP, with the primary goal of expanding the objective function with respect to the optimization variables in subdomains of the original search space. The resulting algorithm, referred to as GASP–DA, is summarized in the following for the P_1–P_2 transfer problem (8.16):

1. Subdivide the search space $\mathbf{x} = \{\ T_1,\ t_{12}\}$ into boxes and put them in a list $\mathcal{L}$.
2. While $\mathcal{L} \neq \phi$,
    - i. Take out a box $\mathbf{X}$ from $\mathcal{L}$.
    - ii. Initialize T_1 and t_{12} as DA variables and compute the Taylor expansion of ΔV_1 on $\mathbf{X}$.
    - iii. Bound the polynomial expansion of ΔV_1 on $\mathbf{X}$, i.e., estimate its minimum $\underline{\Delta V_1}$ and maximum $\overline{\Delta V_1}$ on $\mathbf{X}$.
    - iv. If $\underline{\Delta V_1} > \Delta V_1^{max} \Rightarrow$ discard the current box $\mathbf{X}$ and go to step *i*.
    - v. Compute the Taylor expansion of ΔV_2 on $\mathbf{X}$.
    - vi. Bound the polynomial expansion of ΔV_2 on $\mathbf{X}$, i.e., estimate its minimum $\underline{\Delta V_2}$ and maximum $\overline{\Delta V_2}$ on $\mathbf{X}$.
    - vii. If $\underline{\Delta V_2} > \Delta V_2^{max} \Rightarrow$ discard the current box $\mathbf{X}$ and go to step *i*.
    - viii. Put $\mathbf{X}$ in a list of feasible boxes $\mathcal{X}$.

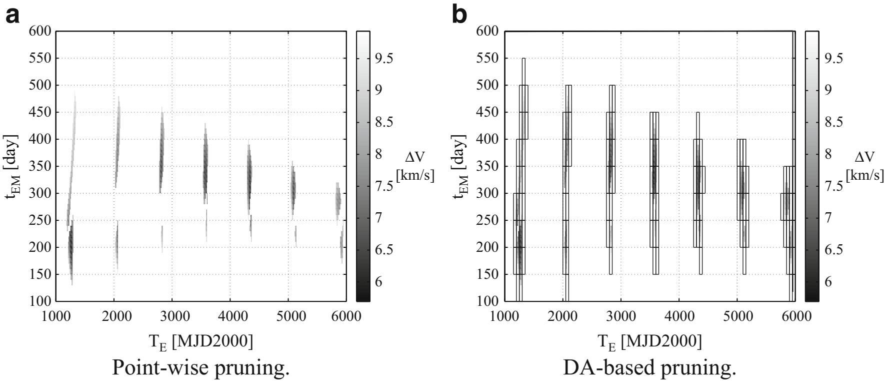


Fig. 8.6 Search space pruning for Earth–Mars transfers

It is worth mentioning that bounding the Taylor expansions, as required in steps 2.iii and 2.vi of the previous algorithm, is not a trivial task. This is done with a non-validated quadratic bounder [17]. The bounder makes use of the quadratic part of the Taylor expansion to get estimates of the minimum of a function over each box.

To assess the performances of GASP-DA, its application to an Earth–Mars transfer is analyzed. A search space of 5,000 days on the departure epoch ($T_1 \in$ [1000, 6000] MJD2000) and 500 days on the transfer time ($t_{12} \in$ [100, 600]) is selected. Figure 8.6a is obtained with classical pointwise techniques, and reports the search space remaining after imposing the two constraints:

$$\begin{aligned} \Delta V_1 &\leq 5\ km/s, \\ \Delta V_2 &\leq 5\ km/s. \end{aligned} \tag{8.28}$$

In the DA implementation of problem (8.16), the search space is uniformly subdivided in boxes of size 50 days on each variable, and the pruning is then performed using the constraints (8.28). The boxes remaining after pruning (Fig. 8.6b) sharply enclose the feasible space in Fig. 8.6a. A comprehensive assessment of the performances of GASP-DA can be found in [9].

8.4 Introduction of Deep Space Maneuvers in GASP-DA

The GASP-DA algorithm is extended in this section to manage DSM. These maneuvers are usually carried out to improve the performances of the transfer trajectories in terms of total cost. From the trajectory optimization standpoint, the introduction of DSM increases the chances of reducing the overall transfer cost associated to pure MGA transfers. On the other hand, each DSM involves additional degrees of freedom that widen the search space and affect convergence to the global minimum.

The mathematical formulation of this new problem is not unique, and the performances of the optimization process strongly depend upon problem transcription, especially in the DA frame. Different formulations have been investigated by the authors in [17]. After some preliminary considerations on the introduction of DSM in MGA transfers, this section describes the strategy that better fits the DA implementation of GASP. The performances of the resulting GASP–DSM–DA algorithm are then assessed on practical cases.

8.4.1 Preliminary Considerations

The solution of several Lambert's problems is yet at the basis of the objective function evaluation in an MGA-DSM problem. However, unlike MGA problems, Lambert's arcs connect either two consecutive planets, or a planet to a maneuver point (and vice versa). The location of the maneuver points has to be specified by adding new variables to the decision vector. It can be easily shown that, for each DSM introduced, a minimum set of four variables must be added for a three-dimensional transfer problem (three variables in the planar case). Based on rationales in [9, 17] the search space pruning of MGA-DSM transfers is carried out in a planar model. The optimal, spatial trajectory is then caught by the subsequent optimization in the three-dimensional environment. This implies that, letting n_{P} and n_{D} be the number of planets and maneuvers, respectively, the decision vector for search space pruning includes $n_{\mathrm{P}} + 3n_{\mathrm{D}}$ variables.

Similarly to the MGA case, the pruning process of MGA-DSM problems consists in (1) expanding the objective function and constraints in Taylor series of the decision variables over subsets of the search space, (2) bounding the resulting polynomials and (3) pruning away unfeasible boxes from the search space. Unlike a point-wise approach, the performance of the whole procedure depends on the availability of accurate range bounds of the constraint functions over each box. Thus, working with smooth functions of the least number of variables is desirable to efficiently prune the search space. Different strategies for the introduction of DSM show different dependencies on the decision variables, which is the key aspect in a DA framework.

An additional consideration concerns the increased computational burden when moving from the MGA to the MGA-DSM problem. This pertains not only the increased dimension of the search space (from n_{P} to $n_{\mathrm{P}} + 3n_{\mathrm{D}}$), but rather it is an intrinsic consequence of representing a function with its Taylor expansion. The number of monomials needed to represent a function of v variables up to the order n is $NM = (k + v)!/(k!v!)$. Thus, at fixed k, the number of monomials for a MGA-DSM problem increases with factorial law with respect to a simple MGA problem, together with the number of required operations.

Based on the previous observations, the complexity of MGA problems increases when DSM are introduced. Nevertheless, the associated issues can be prevented and limited by carefully selecting the strategy for DSM introduction. It is anyway

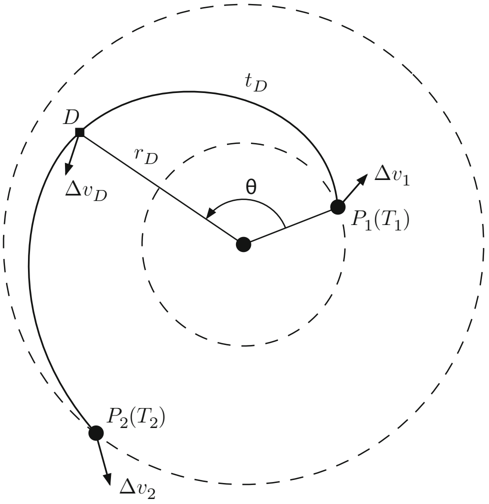


Fig. 8.7 Planet-to-planet transfer with one DSM

important to preserve the idea of the GASP algorithm: subdivide the problem into a cascade of subproblems and exploit the cut-off values to prune away unfeasible zones.

8.4.2 Formulation of GASP–DSM–DA

The strategy to introduce DSM into GASP-DA is addressed in the following. The main idea is to identify the problem formulation that most verges the objective function evaluation to the solution of multiple Lambert's problems by breaking the whole transfer trajectory into subsequent Lambert's arcs. The strategy is first illustrated on a simple planet-to-planet transfer. Then, the extension to general MGA transfers is addressed.

8.4.2.1 Planet-to-Planet Case

Let us consider a planar planet-to-planet transfer as defined in Sect. 8.3.2, and let us introduce one intermediate maneuver (D). With reference to Fig. 8.7, three variables are added to the design vector:

- r_D—*maneuver radius* is the distance between D and the Sun
- t_D—*partial tof* is the time of flight associated to the P_1–D arc
- θ—*incremental anomaly* is the anomaly of D relative to P_1

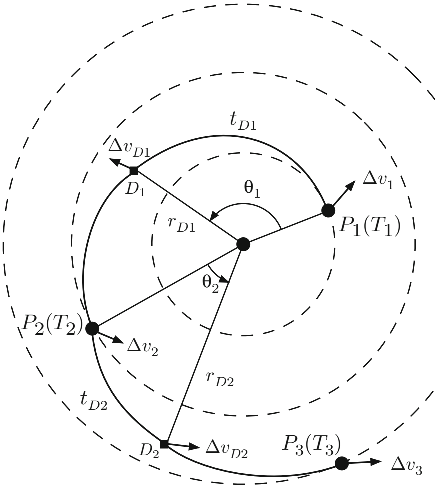


Fig. 8.8 MGA transfer with two DSM

Clearly, $t_D \leq T_2 - T_1$. The ephemeris model gives the position of P_1 at T_1, $\mathbf{r}_1$, and P_2 at T_2, $\mathbf{r}_2$. Thus, the position of D, $\mathbf{r}_D$, is uniquely determined by the angle θ and r_D. Within this strategy, the following dependency holds

$$\mathbf{r}_D = \mathbf{r}_D(T_1, r_D, \theta).$$

The overall transfer can be characterized by solving two Lambert's problems: one from $\mathbf{r}_1$ to $\mathbf{r}_D$ with time of flight t_D and one from $\mathbf{r}_D$ to $\mathbf{r}_2$ with time of flight $T_2 - T_1 - t_D$. Thus, the decision vector is $\mathbf{x} = [T_1, T_2, r_D, \theta, t_D]$.

The search space pruning problem consists now in finding $\mathscr{X}$ such that

$$\exists\ \mathbf{x}^* \in \mathscr{X} \mid \ \Delta v_1(\mathbf{x}^*) \leq \Delta v_1^{max}, \ \ \Delta v_D(\mathbf{x}^*) \leq \Delta v_D^{max}, \ \ \Delta v_2(\mathbf{x}^*) \leq \Delta v_2^{max}, \qquad (8.29)$$

where Δv_D is the cost of the DSM and $\Delta v_D{}^{\max}$ is its maximum allowed value. The dependencies of the three functions in Eq. (8.29) are

$$\begin{aligned} \Delta v_1 &= \Delta v_1(T_1, r_D, \theta, t_D), \\ \Delta v_D &= \Delta v_D(T_1, T_2, r_D, \theta, t_D), \\ \Delta v_2 &= \Delta v_2(T_1, T_2, r_D, \theta, t_D). \end{aligned} \qquad (8.30)$$

Thus, all constraint functions depend on five variables at most.

8.4.2.2 MGA Case

The MGA transfer case is now addressed. Referring to Fig. 8.8, we first consider a MGA case with three planets (P_1, P_2, P_3) and two DSM (D_1 and D_2). The search space is defined by the decision vector

$$\mathbf{x} = [T_1, T_2, T_3, r_{D_1}, \theta_1, t_{D_1}, r_{D_2}, \theta_2, t_{D_2}],$$

where the last three variables are introduced to identify the position of the second DSM, $\mathbf{r}_{D_2}$, and the transfer time between planet P_2 and D_2, t_{D_2}. The inequality $t_{D_1} \leq T_2 - T_1$ still holds, whereas t_{D_2} is subject to $t_{D_2} \leq T_3 - T_2$. Thus, the overall transfer can be characterized by solving four Lambert's problems. The pruning problem consists in finding $\mathscr{X}$ such that $\exists\ \mathbf{x}^* \in \mathscr{X}$ that yields

$$\begin{gathered}\Delta v_1(\mathbf{x}^*) \leq \Delta v_1^{max}, \Delta v_{D_1}(\mathbf{x}^*) \leq \Delta v_{D_1}^{max}, \Delta v_2(\mathbf{x}^*) \leq \Delta v_2^{max},\\ r_p(\mathbf{x}^*) \geq r_p^{\min}, \Delta v_{D_2}(\mathbf{x}^*) \leq \Delta v_{D_2}^{max}, \Delta v_3(\mathbf{x}^*) \leq \Delta v_3^{max},\end{gathered} \tag{8.31}$$

where ${r_p}^{\min}$ is the minimum allowed pericenter radius for the gravity assist at P_2.

Analogously to the previous case, we are interested in assessing the dependence of the position of D_1 and D_2, $\mathbf{r}_{D_1}$ and $\mathbf{r}_{D_2}$, on the problem variables. The main advantage of the approach proposed is that $\mathbf{r}_{D_2}$ is identified on the basis of the position of P_2, and therefore

$$\begin{aligned}\mathbf{r}_{D_1} &= \mathbf{r}_{D_1}(T_1, r_{D_1}, \theta_1),\\ \mathbf{r}_{D_2} &= \mathbf{r}_{D_2}(T_2, r_{D_2}, \theta_2).\end{aligned}$$

Consequently, after the solution of the Lambert problems, the constraint functions show the dependencies

$$\begin{aligned}\Delta v_1 &= \Delta v_1(T_1, r_{D_1}, \theta_1, t_{D_1}),\\ \Delta v_{D_1} &= \Delta v_{D_1}(T_1, T_2, r_{D_1}, \theta_1, t_{D_1}),\\ \Delta v_2 &= \Delta v_2(T_1, T_2, r_{D_1}, \theta_1, t_{D_1}, r_{D_2}, \theta_2, t_{D_2}),\\ r_p &= r_p(T_1, T_2, r_{D_1}, \theta_1, t_{D_1}, r_{D_2}, \theta_2, t_{D_2}),\\ \Delta v_{D_2} &= \Delta v_{D_2}(T_2, T_3, r_{D_2}, \theta_2, t_{D_2}),\\ \Delta v_3 &= \Delta v_3(T_2, T_3, r_{D_2}, \theta_2, t_{D_2}).\end{aligned} \tag{8.32}$$

As can be seen, the critical functions in terms of dependencies are $\varDelta v_2$ and r_p, which depend on eight variables. These are the constraint functions associated to the gravity assist at P_2, which is located between D_1 and D_2.

From simple reasoning, this result can be extended to a general MGA transfer problem with at most one DSM within each planet-to-planet arc. More specifically, let us consider a MGA transfer with n planets $P_1, \ldots, P_i, \ldots, P_n$ and $n - 1$ maneuvers $D_1, \ldots, D_i, \ldots, D_{n-1}$, where D_i is performed between P_i and P_{i+1}. The dependency of constraint functions is

$$\begin{aligned}\Delta v_1 &= \Delta v_1(T_1, r_{D_1}, \theta_1, t_{D_1}),\\ \Delta v_{D_{i-1}} &= \Delta v_{D_{i-1}}(T_{i-1}, T_i, r_{D_{i-1}}, \theta_{i-1}, t_{D_{i-1}}),\\ \Delta v_i &= \Delta v_i(T_{i-1}, T_i, r_{D_{i-1}}, \theta_{i-1}, t_{D_{i-1}}, r_{D_i}, \theta_i, t_{D_i}),\\ r_{p_i} &= r_{p_i}(T_{i-1}, T_i, r_{D_{i-1}}, \theta_{i-1}, t_{D_{i-1}}, r_{D_i}, \theta_i, t_{D_i}),\\ \Delta v_n &= \Delta v_n(T_{n-1}, T_n, r_{D_{n-1}}, \theta_{n-1}, t_{D_{n-1}}).\end{aligned} \tag{8.33}$$

for $i = 2, \ldots, n - 1$, where r_{p_i} is the pericenter radius of the gravity assist at P_i. As can be seen, the proposed strategy limits the maximum dependency to eight variables, regardless of the number of planets and maneuvers.

8.5 Test Cases

A number of application cases are dealt with in this section to assess the performances of the algorithm. More specifically, the classic Earth–Mars transfer, with an intermediate DSM, is first discussed (Sect. 8.5.2). After this simple case, four transfer options for a mission to Jupiter are taken into account (Sects. 8.5.3–8.5.6). These cases differ in the transfer strategy, although they all include only one DSM. The last two cases are devoted to Cassini-like transfers with one and two DSM (Sects. 8.5.7 and 8.5.8, respectively). For each case, the global optimum achieved is compared with the results of GASP-DA, where pure MGA transfers are treated [9].

The outcome of the pruning process is a list of boxes that enclose feasible regions of the search space. A local optimization is then carried out within the remaining boxes to locate the minimum of the objective function, which is the purpose of the original optimization problem. This choice speeds-up the local optimization as the optimizer runs over small domains. Thus, the whole pruning and optimization sequence is implemented in a deterministic way, and the repeatability of the results is preserved. The computational time is relative to a PC with 2.01 Ghz CPU and 512 Mb RAM.

8.5.1 Search Space Definition

The search space bounds and the size of the boxes in which it is subdivided are chosen depending on the problem to solve. This is valid for both the n_{P} epochs and the $3n_{\mathrm{D}}$ auxiliary variables that identify the DSM. The bounds and box-size for the n_{P} epochs are given in dedicated tables. Specific arguments must be provided for the selection of bounds and box-sizes for the $3n_{\mathrm{D}}$ auxiliary variables. These values, reported in Table 8.1, have been selected heuristically, based on the results of an extensive test campaign. They represent a good trade-off between the accuracy of the Taylor representation on the resulting boxes and a limited computational time. The table shows the lower and upper bounds for these variables. These bounds are relative to a maneuver located in the transfer arc between P_i and P_{i+1} (Fig. 8.7). The bounds for θ are trivial. The terms r_i and r_{i+1}, $r_{i+1} > r_i$, stand for the mean radii of P_i and P_{i+1} orbits, respectively.

Table 8.1 Bounds and box-sizes for the $3n_{\mathrm{D}}$ auxiliary variables

Variable	Lower bound	Upper bound	Box-size
r_D	$0.\ 9r_i$	$1.\ 1r_{i\ +\ 1}$	0.1 AU
θ	0	2π	10 deg
t_D	0	$T_{i+1} - T_i$	50 day

Table 8.2 Search space bounds, box-sizes, and optimal solution found for the EdM transfer

	T_{E}	t_{EM}
	MJD 2000	days
L_b	1,000	200
U_b	2,000	650
Δ	50	50
Sol.	1,243.2	606.2

Thus, the maneuver is constrained to lie into an annular region enclosing the planets orbits (If $r_i > r_{i\ +\ 1}$, then $r_D \in [r_{i\ +\ 1}, r_i]$). When the maneuver is located between two encounters of the same planet, i.e., $P_i = P_{i+1}$, we set $r_D \in [0.\ 9r_i$, $1.\ 1a_{1:\ 2}]$, where $a_{1:\ 2}$ is the semimajor axis of an orbit in 1:2 resonance with the orbit of P_i. We let the partial time of flight, t_D, to vary within $[0, T_{i+1} - T_i]$, where T_i and $T_{i\ +\ 1}$ are the epochs at the P_i and $P_{i\ +\ 1}$ encounter, respectively.

8.5.2 EdM

The first test case is an Earth–Mars transfer with one DSM (indicated with *d* in the planets sequence). The search space is defined in Table 8.2 in terms of bounds on the departure epoch T_{E} and the transfer time t_{EM}. However, as stated in the previous sections, the pruning is carried out on the search space defined by the epochs of each planet encounter, i.e., T_{E} and T_{M} in this case. This observation holds for all test cases. The search space definition is completed by the bounds for the three additional variables in Table 8.1. The last two rows of Table 8.2 report the box-size along each epoch and the optimal solution found, respectively. The GASP–DSM–DA algorithm solves this problem in 253.2 s. Wesummarize below some features of the problem settings and the resultsachieved.

- Problem constraints: $\Delta v_{\mathrm{E}} \leq 3$ km/s, $\Delta v_d \leq 3$ km/s, $\Delta v_{\mathrm{M}} \leq 3$ km/s, $\Delta v_{\mathrm{tot}} \leq 7\{\mathrm{km/s}\}$
- Total number of boxes = 388,800
- Boxes remaining after pruning = 1,603 (0.41%)
- Optimal objective function value = 5.632 km/s
- Optimal objective function value without DSM (GASP-DA result) = 5.667 km/s

Table 8.3 Search space bounds, box-sizes, and optimal solution found for the EMdJ transfer

	T_E	t_{EM}	t_{MJ}
	MJD 2000	days	days
L_b	1,000	300	1,000
U_b	3,000	700	2,000
Δ	50	50	100
Sol.	2,804.7	321.9	1,161.9

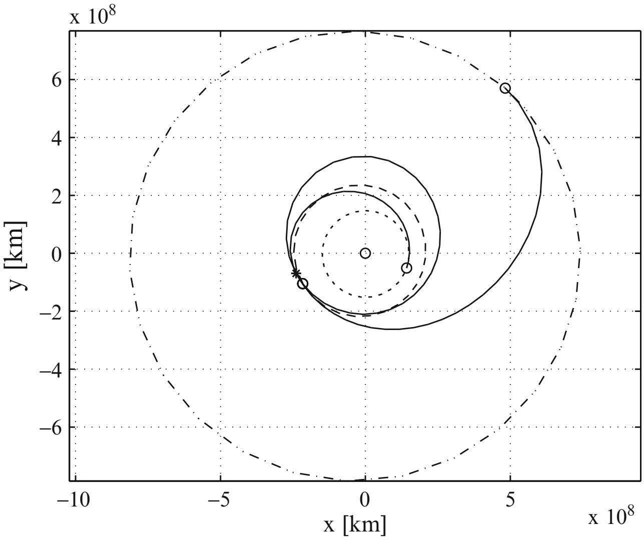


Fig. 8.9 Optimal EMdJ transfer

8.5.3 EMdJ

One planet is added to the transfer. In particular, an Earth–Mars–Jupiter transfer is investigated, with one DSM between Mars and Jupiter. Table 8.3 states the bounds and the box-sizes on the departure epoch and the transfer times, together with the optimal solution found. A purely ballistic Mars gravity assist is imposed by setting the cutoff value for Δv_M to zero in the powered gravity assist model. As in the previous problem, the introduction of a DSM improves the objective function found by GASP-DA. The CPU time is 451 s. The optimal transfer is shown in Fig. 8.9.

- Constraints: $\Delta v_E \leq 5$ km/s, $\Delta v_M \leq 0$ km/s, $\Delta v_d \leq 5$ km/s, $\Delta v_J \leq 5$ km/s, $\Delta v_{tot} \leq$ 15 km/s
- Total number of boxes = 8.52e7
- Boxes remaining after pruning = 323 (3.79e-4%)
- Optimal objective function value = 12.481 km/s
- Optimal objective function value without DSM (GASP-DA result) =13.416 km/s

Table 8.4 Search space bounds, box-sizes, and optimal solution found for the EMdMJ transfer

	T_E	t_{EM}	t_{MM}	t_{MJ}
	MJD 2000	days	days	days
L_b	3,650	30	330	600
U_b	7,300	430	830	2,000
Δ	50	100	100	200
Sol.	4,353.8	371.1	915.8	1,129.5

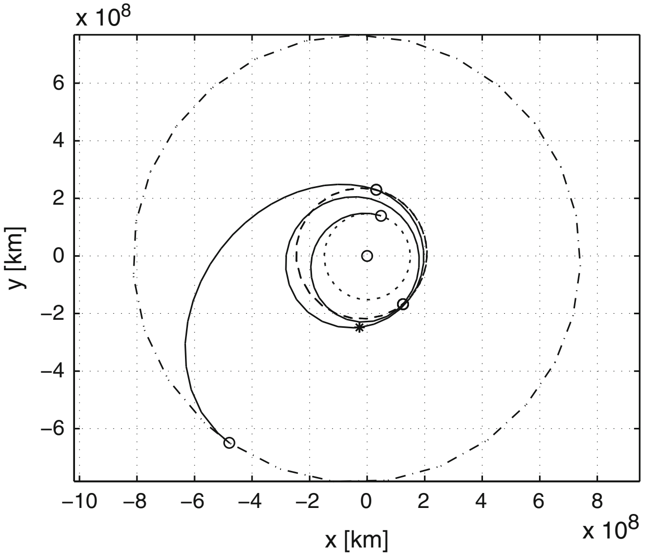


Fig. 8.10 Optimal EMdMJ transfer

8.5.4 EMdMJ

An alternative transfer strategy to Jupiter is investigated. The time domain for the EMdMJ problem is stated in Table 8.4. In this case the maneuver radius is search within $r_D \in [0.\ 9r_M, 1.\ 1a_{1:\ 2}]$, where r_M is the mean radius of Mars' orbit whereas $a_{1:\ 2}$ is the semimajor axis of a 1:2 resonant orbit with Mars' orbit. This problem is solved in 144.2 s. The result obtained by GASP-DA for the pure MGA transfer without DSM is once again improved. Figure 8.10 illustrates the resulting optimal transfer.

- Constraints: $\Delta v_E \leq 4$ km/s, $\Delta v_{M,1} \leq 0$ km/s, $\Delta v_d \leq 3$ km/s, $\Delta v_{M,2} \leq 0$ km/s, $\Delta v_J \leq 7$ km/s, $\Delta v_{tot} \leq 12$ km/s
- Total number of boxes = 9.19e7
- Boxes remaining after pruning = 717 (7.8e-3%)
- Optimal objective function value = 10.843 km/s
- Optimal objective function value without DSM (GASP-DA result) = 12.864 km/s

Table 8.5 Search space bounds, box-sizes, and optimal solution found for the EVdVEJ transfer

	T_{E}	t_{EV}	t_{VV}	t_{VE}	t_{EJ}
	MJD 2000	days	days	days	days
L_b	3,650	80	80	80	600
U_b	7,300	430	830	830	2,000
$\mathit{\Delta}$	50	25	25	50	200
Sol.	3,859.5	119.2	429.6	564.8	1,244.3

Table 8.6 Search space bounds, box-sizes, and optimal solution found for the EVEdEJ transfer

	T_{E}	t_{EV}	t_{VE}	t_{EE}	t_{EJ}
	MJD 2000	days	days	days	days
L_b	3,650	80	80	80	600
U_b	7,300	430	830	830	2,000
$\mathit{\Delta}$	50	25	50	50	200
Sol.	3,863.4	128.8	288.4	713.3	1,068.2

8.5.5 EVdVEJ

The MGA transfer EVdVEJ to Jupiter is now analyzed. One DSM maneuver is performed between the two consecutive Venus encounters. The five-dimensional domain for the departure epoch and the transfer times is defined in Table 8.5. The domain for DSM characterization is added based on Table 8.1. The resulting search space is relatively large, and 1.85e10 boxes are necessary to run GASP–DSM–DA. Thanks to constraint propagation, this problem is solved in 2,770 s.

- Constraints: $\mathit{\Delta} v_{\mathrm{E,dep}} \leq 4.5\,\mathrm{km/s}$, $\mathit{\Delta} v_{\mathrm{V},1} \leq 0\,\mathrm{km/s}$, $\mathit{\Delta} v_d \leq 0.5\,\mathrm{km/s}$, $\mathit{\Delta} v_{\mathrm{V},2} \leq 0\,\mathrm{km/s}$, $\mathit{\Delta} v_{\mathrm{E}} \leq 0$ km/s, $\mathit{\Delta} v_{\mathrm{J}} \leq 7$ km/s, $\mathit{\Delta} v_{\mathrm{tot}} \leq 12$ km/s
- Total number of boxes = 1.85e10
- Boxes remaining after pruning = 3.80e4 $(2.06\mathrm{e} - 4\%)$
- Optimal objective function value = 9.304 km/s
- Optimal objective function value without DSM (reference solution [18]) = 10.503 km/s

8.5.6 EVEdEJ

The last transfer strategy to Jupiter is now studied. An EVEdEJ transfer problem is solved using the search space bounds and the box-sizes reported in Table 8.6. Similarly to the previous problem, the search space is relatively large, and a systematic analysis based on a grid sampling would be impossible without an efficient constraint propagation. Thanks to the pruning strategy and the possibility of expanding the constraint functions over subdomains of the search space, GASP–DSM–DA solves this problem in 2,392 s. The main results are listed below, and a plot of the optimal transfer is reported in Fig. 8.11.

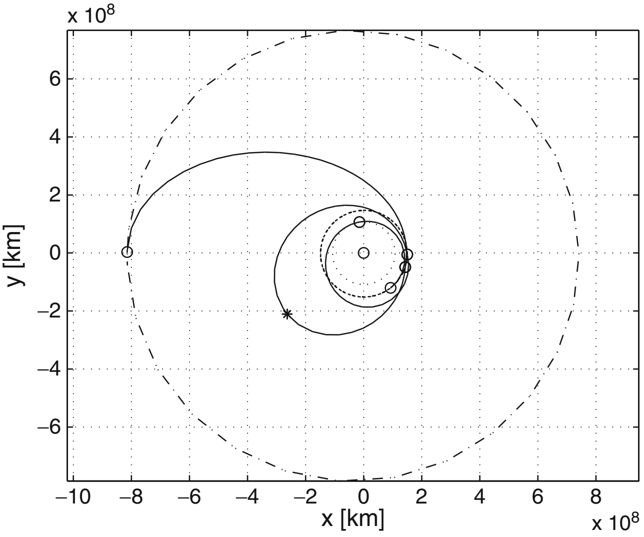


Fig. 8.11 Optimal EVEdEJ transfer

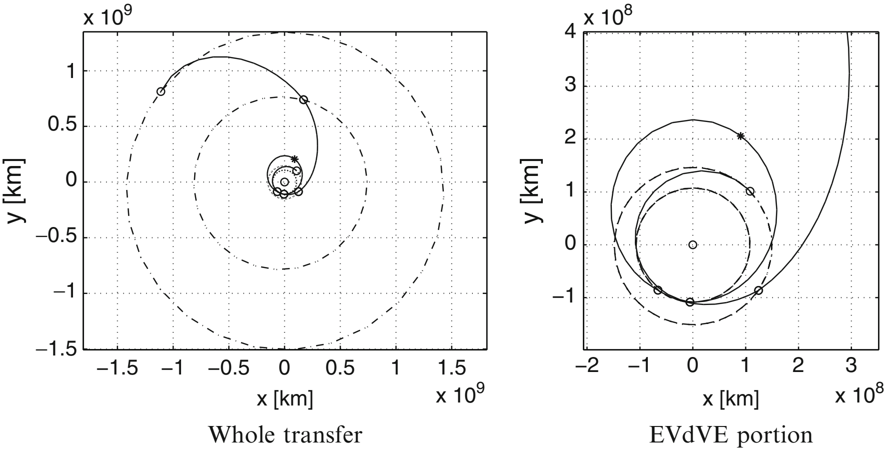
Table 8.7 Search space bounds, box-sizes, and optimal solution found for the EVdVEJS transfer

	T_E	t_{EV}	t_{VV}	t_{VE}	t_{EJ}	t_{JS}
	MJD 2000	days	days	days	days	days
L_b	− 1,000	80	200	30	400	800
U_b	0	430	500	180	1,600	2,200
$\varDelta$	50	25	25	50	200	200
Sol.	− 787.0	165.8	427.7	57.7	596.1	2,200

- Constraints: $\varDelta v_{\mathrm{E,dep}} \leq 4$ km/s, $\varDelta v_{\mathrm{V}} \leq 0$ km/s, $\varDelta v_{\mathrm{E,1}} \leq 0$ km/s, $\varDelta v_d \leq 3$ km/s, $\varDelta v_{\mathrm{E,2}} \leq 0$ km/s, $\varDelta v_{\mathrm{J}} \leq 7$ km/s, $\varDelta v_{\mathrm{tot}} \leq 12$ km/s
- Total number of boxes = 9.25e9
- Boxes remaining after pruning = 4.84e4 (5.23e − 4%)
- Optimal objective function value = 8.670 km/s
- Optimal objective function value without DSM (GASP-DA result) = 10.09 km/s; reference solution [18] = 8.680 km/s

8.5.7 EVdVEJS

This section is devoted to a Cassini-like transfer with a DSM between the two Venus gravity assists. Saturn is the target planet, which is reached after four gravity assists. Thus, the overall transfer involves six planet encounters and one DSM, leading to a nine-dimensional optimization problem. The search space and the box-sizes are stated in Table 8.7. The computational time required by the DA-based



Whole transfer

EVdVE portion

Fig. 8.12 Optimal EVdVEJS transfer

pruning and optimization algorithm is 210 s. Figure 8.12a illustrates the optimal transfer, whereas a detail on the EVdVE portion is reported in Fig. 8.12b.

- Constraints: $\Delta v_{\mathrm{E,dep}} \leq 4$ km/s, $\Delta v_{\mathrm{V},1} \leq 1$ km/s, $\Delta v_d \leq 1$ km/s, $\Delta v_{\mathrm{V},2} \leq 0$ km/s, $\Delta v_{\mathrm{E}} \leq 0$ km/s, $\Delta v_{\mathrm{J}} \leq 0$ km/s, $\Delta v_{\mathrm{S}} \leq 5$ km/s
- Total number of boxes = 3.92e8
- Boxes remaining after pruning = 2,281 (1e − 3%)
- Optimal objective function value = 8.299 km/s
- Optimal objective function value without DSM (GASP-DA result) = 8.619 km/s

8.5.8 EVdVEJdS

An additional DSM is now introduced in the Jupiter–Saturn arc of the Cassini-like transfer of Sect. 8.5.7. The search space is analogous to that reported in Table 8.7, except for t_{JS}, which ranges from 1,600 to 3,000 days. The GASP–DSM–DA algorithm prunes efficiently the twelve-dimensional search space to 1e − 6% of the initial size. The CPU time is 2,000 s. The main results of this problem are listed below.

- Constraints: $\Delta v_{\mathrm{E,dep}} \leq 4$ km/s, $\Delta v_{\mathrm{V},1} \leq 1$ km/s, $\Delta v_{d,\,1} \leq 1$ km/s, $\Delta v_{\mathrm{V},2} \leq 0$ km/s, $\Delta v_{\mathrm{E}} \leq 0$ km/s, $\Delta v_{\mathrm{J}} \leq 0$ km/s, $\Delta v_{d,\,2} \leq 1$ km/s, $\Delta v_{\mathrm{S}} \leq 5$ km/s
- Total number of boxes = 1.41e12
- Boxes remaining after pruning = 2.23e4 (1.6e − 6%)
- Optimal objective function value = 8.276 km/s
- Optimal objective function value without DSM (GASP-DA result) = 8.619 km/s

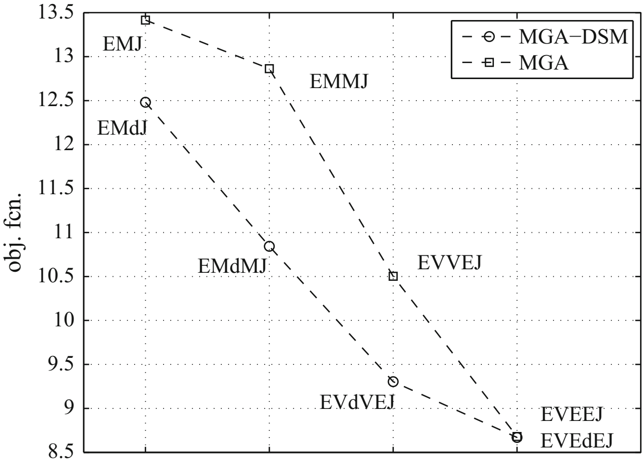


Fig. 8.13 Options for a transfer to Jupiter. Both the optimal MGA (*squares*) and MGA-DSM (*circles*) solutions are shown

8.6 Final Remarks

This chapter described how DSM can be inserted in the search space pruning process of the algorithm GASP. The proposed algorithm takes advantage of differential algebra, which is used to expand the constraint functions in Taylor series of the design variables. The adopted problem formulation limits the maximum functional dependency to eight variables, which is important to accurately bound the constraint functions. The resulting GASP–DSM–DA algorithm has been tested to solve relevant interplanetary transfer problems.

The introduction of the DSM into an MGA transfer deserves a final comment. Figure 8.13 compares the optimal objective function values for a mission to Jupiter, obtained with different transfer strategies. More specifically, GASP-DA and GASP–DSM–DA are used to compute the optimal solutions for pure MGA and MGA transfers with one DSM, respectively. Evidently, different transfer strategies have different costs. It is worth noting that, for the cases presented in Fig. 8.13, and generally for the MGA-DSM transfers, the introduction of DSM improves the optimal solutions in terms of transfer cost. On the other hand, MGA-DSM transfers involve longer overall transfer time.

Acknowledgements This research has been carried out under European space agency (ESA)/ Ariadna scheme, contract number: 20271/06/NL/HI. The authors would like to acknowledge the support of Tamás Vinkó and Dario Izzo from the Advanced Concepts Team of ESA. The authors are also grateful to Kyoko Makino for the support and help.

References

1. Yao, X.: Global optimization by evolutionary algorithms. In: Proceeding Of the Second Aizu International Symposium on Parallel Algorithm Architecture Synthesis, Aizu-Wakamatsu, Japan, IEEE Computer Society Press, pp. 282–291 (1997)
2. Ingberg, L.: Simulated annealing: Practice versus theory. Mathl. Comput. Model. **18**, 29–57 (1993)
3. Sentinella, M.R., Casalino, L.: Cooperative evolutionary algorithm for space trajectory optimization. Celestial Mech. Dyn. Astron. **105** (2009). DOI 10.1007/s10569-009-9223-4
4. Jones, D.R., Perttunen, C.D., Stuckman, B.E.: Lipschitzian optimization without the Lipschitz constant. J. Optim. Theory Appl. **79**, 157–181 (1993)
5. Jones, D.R.: A taxonomy of global optimisation methods based on response surfaces. J. Global Optim. **21**, 345–383 (2001)
6. Vasile, M., Summerer, L., De Pascale, P.: Design of Earth–Mars transfer trajectories using evolutionary-branching technique. Acta Astronautica **56**, 705–720 (2005)
7. Myatt, D., Becera, V., Nasuto, S., Bishop, J.: Advanced global optimisation for mission analysis and design. Final Report, Ariadna id: 03/4101, Contract No. 18138/04/NL/MV (2004)
8. Izzo, D., Becerra, V., Myatt, D., Nasuto S., Bishop, J.: Search space pruning and global optimisation of multiple gravity assist spacecraft trajectories. J. Global Optim. **38**, 283–296 (2006)
9. Armellin, R., Di Lizia, P., Topputo, F., Lavagna, M., Bernelli-Zazzera, F., Berz, M.: Gravity assist space pruning based on differential algebra. Celestial Mech. Dynam. Astron. **106**, 1–24 (2010)
10. Berz, M.: Modern Map Methods in Particle Beam Physics. Academic, New York (1999)
11. Berz, M.: The new method of TPSA algebra for the description of beam dynamics to high-orders. Technical Report AT-6:ATN-86-16, Los Alamos National Laboratory (1986)
12. Berz, M.: The method of power series tracking for the mathematical description of beam dynamics. Nucl. Instrum. Meth. **A258**, 431–436 (1987)
13. Berz, M.: Differential Algebraic Techniques. In: Entry in Handbook of Accelerator Physics and Engineering. World Scientific, New York (1999)
14. Berz, M., Makino, K.: COSY INFINITY version 9 reference manual. MSU Report MSUHEP-060803, Michigan State University, East Lansing, MI 48824, 1–84 (2006)
15. Giorgini, J.D., Yeomans, D.K., Chamberlin, A.B.., Chodas, P.W., Jacobson, R.A., Keesey, M.S., Lieske, J.H., Ostro, S.J., Standish, E.M., Wimberly, R.N.: Horizons, JPL's On-Line Solar System Data and Ephemeris Computation Service. User's guide (1998)
16. Battin, R.H.: An Introduction to the Mathematics and Methods of Astrodynamics, 2nd Printing. AIAA Education Series, Providence (1987)
17. Bernelli-Zazzera, F., Berz, M., Lavagna, M., Armellin, R., Di Lizia, P., Topputo, F.: Global Trajectory Optimisation: Can We Prune the Solution Space when Considering Deep Space Maneuvers? Final Report, Ariadna id: 06/4101, Contract No. 2007/06/NL/HI (2006)
18. Vasile, M., De Pascale, P.: Preliminary design of multiple gravity-assist trajectories. J. Spacecraft Rockets **43**, 794–805 (2006)