Spiral Trajectories in Global Optimisation of Interplanetary and Orbital Transfers

Final Report

Authors: M. Vasile, O. Schütze, O. Junge, G. Radice, M. Dellnitz
Affiliation: University of Glasgow, Department of Aerospace Engineering, James Watt South Building, G12 8QQ, Glasgow,UK

ESA Research Fellow/Technical Officer: Dario Izzo

Contacts:

Massimiliano Vasile
Tel: ++44 141-330-6465
Fax: ++44-141-330-5560
e-mail: mvasile@aero.gla.ac.uk

Dario Izzo, Ph.D., MSc
Tel: +31(0)71 565 3511
Fax: +31(0)71 565 8018
e-mail: act@esa.int

Available on the ACT website
http://www.esa.int/act

Ariadna ID: AO4919 05/4106
Study Duration: 4 months
Contract Number: 19699/06/NL/HE

Table of Contents

1 Introduction

NASA's Deep Space 1 and recently ESA's SMART-1 have shown the effectiveness of low-thrust systems as primary propulsion devices. This opens the door to new kinds of missions in the solar system, exploiting the beneficial effects given by the combination of gravity assist maneuvers and low thrust. In recent studies, missions to Mercury, Jupiter and the Sun have been designed, resorting to a combination of low-thrust and multiple swing-bys. Such new scenarios, characterized by new propulsion systems and by highly challenging mission objectives, make the task of mission analysts more difficult than ever. In fact, the design of low-thrust transfers generally requires the solution of an optimal control problem, which has no general solution in closed form. Different methods have been developed to tackle these trajectory design problems and it is common practice to classify them into two large families: direct and indirect methods. The former family collects all those methods that transcribe the infinite dimensional optimal control problem into a finite parameter optimization problem and then solve the resulting system of nonlinear algebraic equations with a general nonlinear programming approach. Several flavors of transcription methods have been devised ranging from shooting techniques, collocation methods, based on Hermite-Simpson formulas, Lobatto quadrature, Finite Elements in Time or multispectral methods to differential inclusions. The latter family collects all those methods that explicitly derive the necessary conditions for optimality as given by optimal control theory. The solution of the resulting differential algebraic system of equations with boundary conditions is then performed through numerical integration and some kind of gradient method.

All of these approaches need to be fed with a first guess solution which, depending on the particular method, may be quite far from the final optimal solution or needs to be very close to it. The generation of a suitable first guess turns out to be a tricky and quite time consuming task since it deeply effects the final result and the convergence of all the above mentioned methods. For this reason all of them turn out to be unsuitable for the preliminary assessment of a large number of solutions as is usually required in pre-phase A studies. The main difficulty comes from the unavailability of sufficiently general analytical solutions; in fact few cases of analytical solutions of a thrusting spacecraft exist in literature and–although very interesting–they are quite specialized (i.e. restricted to two space dimensions or useful only for particular trajectories). For more general cases, numerical propagation is necessary, with a considerable computational effort even in a very preliminary phase. Studies on the characterization of transfer options for low-thrust trajectories and on the generation of first guess solutions date back to the late nineties with the works of Coverstone et al. [4, 26], where multi-objective genetic algorithms were first used to compute first guess solutions for an indirect method.

The derivation of approximate analytical solutions was addressed in the work of Markopoulos [15], Bishop and Azimov [1, 2]. The former has provided a sort of generalization of non-keplerian analytical solutions for the case of a thrusting spacecraft, while the latter two have tackled directly the solution of the optimal control problem for a low-thrust trajectory with very interesting results. Inspired by the work of Tanguay [29], Petropoulos and Longuski [21, 22, 20] proposed a shape-based approach, which represents the trajectory (connecting two points in space) with a particular parameterized analytical curve (or *shape*) and computes the control thrust necessary to satisfy the dynamics. Although the resulting trajectory is not the actual solution of an optimal control problem, by tuning the shaping parameters it is possible to generate solutions which are sufficiently good to be fed into a more detailed optimization process. More precisely, in the work by Petropoulos a thrust arc is represented by an analytical curve, known as exponential sinusoid, which consists of a shape with five free parameters

in polar coordinates. This shape is suitable for the approximation of planar motion and the reduced number of shaping parameters does not allow to satisfy all the possible boundary conditions on position and velocity, while satisfying the constraints on the time of flight and on the level of acceleration; for 3D problems the propellant consumption for out of plane motion is only estimated. By implementing the exponential sinusoid trajectory model in the software code STOUR, Petropoulos and Longuski extended their systematic search for optimal ballistic MGA transfers to the global solution of Low-Thrust Gravity Assist (LTGA) transfers [16, 23, 7].
Recently, it has been shown that whenever a Multiple Gravity Assist (MGA) optimisation problem is characterised by a simple Δv-matching for the swing-bys and no deep-space manoeuvres are present, there exists a polynomial-time algorithm that solves the problem [11, 17]. Namely, a space pruning technique exists with a complexity which is quartic with respect to dimensionality, i.e. in the number of swing-bys, and cubic in the resolution of the discretisation of time variables. If Multi LTGA trajectories (MLTGA) are considered it would be desirable to have an equivalent polynomial-time algorithm. By revisiting Petropoulos' approach, Izzo [10] has derived a form of the exponential-sinusoid which represents the solution of Lambert's problem for low-thrust arcs in two dimensions. This suggests how to embed the solution of Lambert's problem for exponential-sinusoids into a design tool able to solve the MLTGA problem efficiently.

1.1 Study Objectives

In order to address the definition and implementation of an efficient tool for the solution of the MLTGA problem, the present study aims at reaching the following objectives:

I. Implementation of the solution of Lambert's problem for exponential sinusoids in a global optimisation scheme

The primary goal of this study is the implementation and assessment of a preliminary design tool based on the solution of Lambert's problem for exponential sinusoids. If each arc connecting two planets is solved by an exponential sinusoid and the launch, flyby and arrival dates are taken as variables, then the MLTGA problem can be formulated as general global optimisation problem of finite dimension. Notice that if the flyby sequence and the number of spirals for each low thrust arc are kept constant, the problem is perfectly homogeneous since all quantities have real values.
As for the ballistic MGA case, the complexity of the MLTGA case can be studied by looking for the existence of an algorithm that solves the problem in a time which is polynomial in problem dimension (typically the number of flyby bodies). To this aim a pruning algorithm, analogous to the one used for the ballistic MGA case [17], can be applied when the exponential sinusoids are used in addition to ballistic arcs in order to connect the planets. If the problem can be solved in polynomial time then we can say that the problem has polynomial complexity. However, this does not say much about the structure of the solution space: distribution and number of minima, number of nested minima, size and shape of the basins of attraction, etc... Therefore, a preliminary analysis of the structure of the solution space for a simple low thrust direct transfer will be performed by gridding the search space. The result for the low thrust transfer is then compared to a bi-impulsive transfer. Furthermore, a multistart algorithm is applied to both the bi-impulsive and the low thrust transfers. Since multistart is a stochastic algorithm, ones expects that each run converges to a different minimum, depending on the number of initial samples. If the number of minima is small and the initial samples

are sufficiently dense then the algorithm should always converge to the global optimimum. Thus the number of initial samples for the multistart algorithm is increased until the same minimum is found at every run. This preliminary analysis will help to better understand the performance of a general stochastic global optimisation scheme when applied to the MLTGA problem.
In particular, in this study evolutionary strategies (Differential Evolution as proposed by Becerra et al. in previous study [17]) have been used in order to explore the remaining portions of the solution space after the application of the pruning technique. The resulting combination of the exponential sinusoid trajectory model and the global optimisation scheme will be used to characterise complex MLTGA problems (in this sense complexity is a direct function of the number of gravity manoeuvres, sequence of planetary encounters and orbital elements associated to each target celestial body). The application to realistic cases will assess the actual usefulness of this approach for the creation of a preliminary design tool.

II. Assessment of alternative approaches to the preliminary design of LTGA trajectories

Though the combination of the pruning technique and of the exponential sinusoid shaping approach could be a powerful tool for the preliminary design of LTGA trajectories, other techniques are worth of investigation. Three extensions of this combination will be explored:

- The use of a general tool for global single and multiobjective optimization, instead of the dedicated pruning technique.
- The use of a full 3D shaping approach.

These two extensions, while expected to be computationally more expensive, can however provide a wider range of solutions to the MLTGA problem. It would therefore be interesting to investigate if the reward in terms of number and quality of solutions is worth the increase in computational load. In addition, it is to be noted that for low thrust systems the gain in propellant consumption is often completely offset by the cost of operations (which is directly related to the time of flight).
The optimality of an LTGA trajectory is therefore a trade-off between operating time of the engine and the propellant mass consumption. For this reason, a multiobjective approach would be more appropriate and could lead to a better set of solutions for a given MLTGA problem. Among all available global optimization tools, it is proposed to use the set oriented approach implemented in the software package GAIO[1]. Originating in the realm of numerical analysis of dynamical systems, this package implements a hierarchical domain decomposition technique combined with pruning criteria based on dynamical systems ideas for the treatment of single- and multi-objective global optimization problems.
This will represent the frame of reference since it implements a problem independent pruning of the search space coupled with various kinds of sampling. Since GAIO has already been successfully used for several realistic applications, it will serve as a reference for the assessment of any alternative approach to preliminary trajectory design.

III. Application to realistic space mission cases

The above-mentioned algorithms will be tested on a number of realistic cases. In particular, three types of missions will be studied: direct transfers to the asteroid Apophis, an MLTGA transfer to Mercury and an MLTGA transfer to Jupiter.

[1] `http://www.math.upb.de/~agdellnitz/Software/gaio.html`

IV. Assessment of solution optimality and the effectiveness of the proposed approach

The preliminary design approaches investigated in this study will be compared and evaluated in terms of computational performance and quality of the achievable solutions. The computational cost of each approach will be evaluated in terms of the number of function evaluations and the cost for each single evaluation. The quality of the achievable solution is evaluated in terms of its optimality and usefulness at system design level.
Solution optimality can be defined according to a local criterion and a global criterion. The local criterion will be the satisfaction of the necessary and sufficient condition for optimality derived from optimal control theory. This criterion can be applied a posteriori, inserting the obtained control law and the state evolution into the necessary equations for optimality and deriving the Lagrangian multipliers. Once the Lagrangians are available, it is possible to evaluate how far the solution is from satisfying the Pontryaging maximum principle.
The global criterion for optimality will be derived empirically by comparing the solutions obtained with each approach over a given number of runs. The usefulness at system design levels accounts for the trade-offs among several optimality criteria: mass, power, cost and time. Pareto optimality will be used to construct the optimal trade-off curves. Different approaches can be compared in terms of the classical performance metrics, generally used to evaluate multiobjective optimization methods: the distance from the best achieved Pareto set, the spreading of the solutions, and the extension of the Pareto set. In addition, a pair of nondominated sets are compared by calculating the fraction of each set that is covered by the other set. This can be used in order to show that the outcomes of an algorithm dominate the outcomes of another algorithm, although it does not tell how much better it is. In fact, since by construction the set oriented multiobjective optimization schemes (GAIO) always compute a covering of the entire global Pareto set, we will use this scheme as a benchmark for the other approaches. Notice that here the aim is not to test the global optimiser but more to asses the Pareto optimality of the set of solutions that can be obtained with the tested trajectory models and optimisation techniques. As an example, if the exponential sinusoid model is run in conjunction with a pruning technique for a given Earth-GA-NEO sequence, then the solution space will be reduced to a limited number of disconnected sets representing different launch windows. For each launch window a number of Pareto optimal solutions should exist. All these solutions will form a Pareto optimal front for that particular trajectory model and for a given interval of launch dates and transfer times.

2 Trajectory Model and Problem Formulation

In this section we are introducing the trajectory models used for this study. In particular, the exponential sinusoid for low thrust arcs and the powered swing-by model for gravity assist manoeuvres. The problem of finding an optimal MLTGA trajectory will be formulated as a general global optimisation problem.

2.1 The Exponential Sinusoid

It is here proposed to use a particular model for multiple gravity assist low thrust trajectories (MLTGA). Low thrust arcs are modeled through a shape based approach based on the exponential sinusoid proposed by Petropoulos et al.[23]:

$$r = k_0 \exp(k_1 \sin(k_2\theta + \phi))$$

For each transfer arc a two-point baundary value problem (TPBVP) has to be solved. For the solution of the TPBVP we follow the development proposed by Izzo [10]. If the thrust vector is aligned with the velocity, the flight path angle γ and the thrust steering angle α are equal, $\gamma = \alpha$, and the thrust history and the polar angle history are uniquely determined and the control acceleration is given by:

$$F = \frac{\mu}{r^2} \frac{\tan\gamma}{2\cos\gamma} \left[\frac{1}{\tan^2\gamma + k_1 k_2^2 s + 1} - \frac{k_2^2(1 - 2k_1 s)}{(\tan^2\gamma + k_1 k_2^2 s + 1)^2} \right] \tag{2.1}$$

The time variation of the true anomaly is given by:

$$\dot{\theta^2} = \left(\frac{\mu}{r^3}\right) \frac{1}{\tan^2\gamma + k_1 k_2^2 s + 1}, \tag{2.2}$$

with the flight path angle given by:

$$\tan\gamma = k_1 k_2 \cos(k_2\theta + \phi) \tag{2.3}$$

and $s = \sin(k_2\theta + \phi)$. Now by solving the integral:

$$\Delta t = \int \left(\left(\frac{\mu}{r^3}\right) \frac{1}{\tan^2\gamma + k_1 k_2^2 s + 1} \right)^{-1/2} d\theta \tag{2.4}$$

one can compute the actual time of flight.

The exponential sinusoid expresses the variation of the radius as a function of the polar angle θ and depends on three shaping parameters k_0, k_1, k_2 plus a phase parameter ϕ. By fixing the initial and final radius:

$$r_1 = k_0 \exp(k_1 \sin(\phi)) \tag{2.5}$$
$$r_2 = k_0 \exp(k_1 \sin(k_2\bar{\theta} + \phi)) \tag{2.6}$$

two of the three parameters can be computed as a function of the others.
The two position radii can be computed from the ephemeris of two planets or other celestial bodies under consideration. In this case, it is normally required that the transfer trajectory going from one planet to the other is flown in a given time T. This implies that the actual time of flight must be equal to the required time of flight in order to have a physical solution:

$$\Delta t - T = 0. \tag{2.7}$$

If now this time constraint is solved, a third parameter can be determined and the exponential sinusoid becomes a single valued function.
In this form, given the transfer time and the two position vectors at the beginning and at the end of the transfer we can compute the velocities at the two extremal points and the thrust profile. Since only one shaping parameter is free it is not possible to optimise the value of the velocities at the boundaries plus the thrust profile but the problem is equivalent to the Lambert's problem for conic arcs.
Furthermore some analysis reveals that the exponential sinusoid gives physical solutions whenever $k_1 k_2^2 < 1$.

2.2 Gravity Assist Model for the Exponential Sinusoid

Since the values of the velocities at the boundaries are not completely free, given an incoming velocity vector it is not possible in general to match any possible outgoing leg. A match can be obtained through a powered gravity assist manoeuvre. Modeling gravity manoeuvres through powered swing-bys has a very important advantage: it decouples the transfer arcs from each other. Each transfer leg is therefore independent of the others. As was demonstrated in a previous study and as will be demonstrated later in this report, this important property renders the growth of the possible paths polynomial with the number of flybys.
We propose two different models: In the first model, the radius of the flyby pericentre r_p is adjusted until the velocity turning angle constraint is no longer satisfied. A simple impulsive Δv manoeuvre is then inserted at the pericentre in order to match the incoming and the outgoing velocities. In the second, the radius of the flyby pericentre r_p is adjusted until a lower limit is reached. At that point, a Δv manoeuvre is inserted in order to produce an additional turn of the velocity vector and to match the incoming and outgoing velocities.

2.3 Problem Formulation

For $N+1$ celestial bodies, each thrust arc is linked to the others through a sequence of powered swing-bys. Abstractly, we are faced with a problem of the form

$$\begin{aligned} \text{minimise:} \quad & J(y) \\ \text{subject to:} \quad & r_p \geq r_{min}, \end{aligned} \tag{2.8}$$

where the solution vector y is

$$y = [t_0, T_1, k_{2,1}, n_1, ..., T_i, k_{2,i}, n_i, ..., T_N, k_{2,N}, n_N]^T \tag{2.9}$$

and the objective function J is

$$J = 1 - \exp\left(-\frac{\Delta V_{GA} + \Delta V_0}{g_0 I_{sp1}} - \frac{\Delta V_{LT}}{g_0 I_{sp2}}\right), \tag{2.10}$$

where ΔV_{GA} is the sum of all the $\Delta V s$ required to correct every gravity assist manoeuvre, ΔV_0 is the departure manoeuvre, while ΔV_{LT} is the sum of the total ΔV of the low thrust arcs. The two specific impulses I_{sp1} and I_{sp2} are those for a chemical engine and for a low-thrust engine, respectively. In this formulation, the use of low thrust is favorized with respect to the chemical corrections. Given the solution vector y, the values for all the r_p can be computed for each swing-by through a one-dimensional search. To this end, a Newton iteration has been adopted in this study.
The overall process for the composition of an MLTGA trajectory with the exponential sinusoid model can be summarised in the following steps (cf. Fig. 1):

- For each departure date t_0 and set of transfer times T_i
- Compute a exponential sinusoid transfer from A to B
- Compute v_1
- Compute a 2-impulse transfer from B to C
- Compute v_3
- Compute v_2 with pericenter radius r_p
- If $v_2 = v_3$ proceed, otherwise compute matching Dvi at the pericentre of the hyperbola
- Compute the launch impulsive manoeuvre Δv_0
- Compute the arrival impulsive manoeuvre Δv_N
- Compute the sum of all Δv

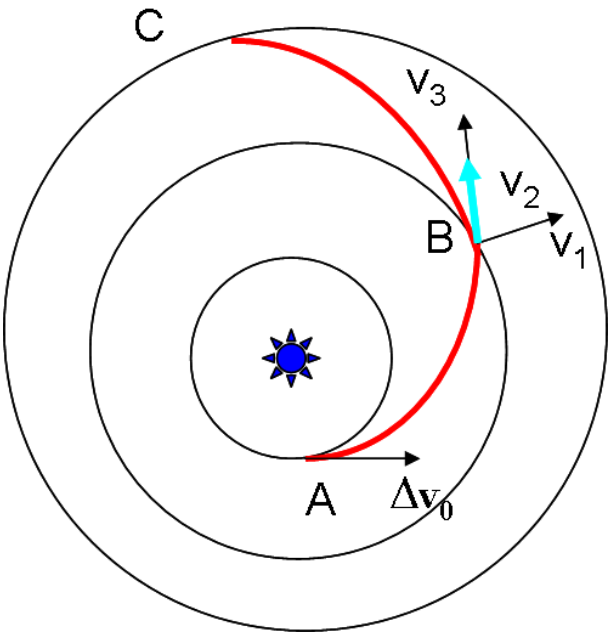


Fig. 1. Composition of a whole MLTGA trajectory for the exponential sinusoid model

3 Problem Analysis

3.1 Search Space Structure

The first step to analyse the problem defined in the previous section was to grid the search space for a direct transfer from the Earth to the astroid Apophis. There are four free variables: the initial time t_0, the time of flight T_0, the k_2 shaping parameter and the number of revolutions n. This four-dimensional space has been analysed fixing the number of revolutions and gridding the other three variables.
The result can be seen in Fig. 2,3 and 4 for different values of k_2 in the range $[0.05, 0.6]$.
If we compare each contour plot with a simple bi-impulsive transfer, as in Fig. 5, we notice that the distribution of minima and the general structure of the search space are very similar. The effect of low-thrust is to 'blur' the basins of attraction of the minima. The general structure and the distribution of the minima remain unchanged also when k_2 changes from 0.05 to 0.275. The white areas on the plot correspond to parts of the search space that are not admissible since the algorithm can not find a solution satisfying the constraint on the time of flight.
It is remarkable that these areas do not correspond only to high values of Δv but also embrace portions of the search space at the border or intersecting the basin of attraction of some of the minima. If one full revolution is inserted before the encounter the whole set of optimal solutions is shifted in time to longer times of flight but the launch windows remain unchanged (cf. Fig. 6,7 and 8). Moreover, a longer transfer time corresponds to an extension of the admissible set though at a value of $k_2 = 0.6$ again part of the search space becomes not admissible.

3.2 Upper limit on k_2

The number of allowable revolutions is dependent on the transfer time and on the value of k_2. For physical reasons, there cannot be more than a certain maximum number of spirals

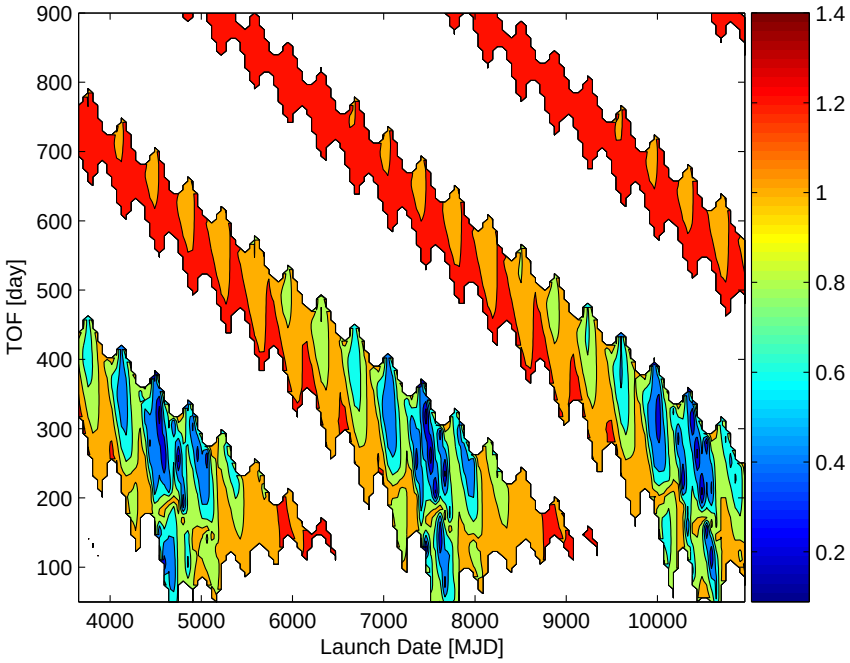


Fig. 2. Objective function J for $k_2 = 0.6$.

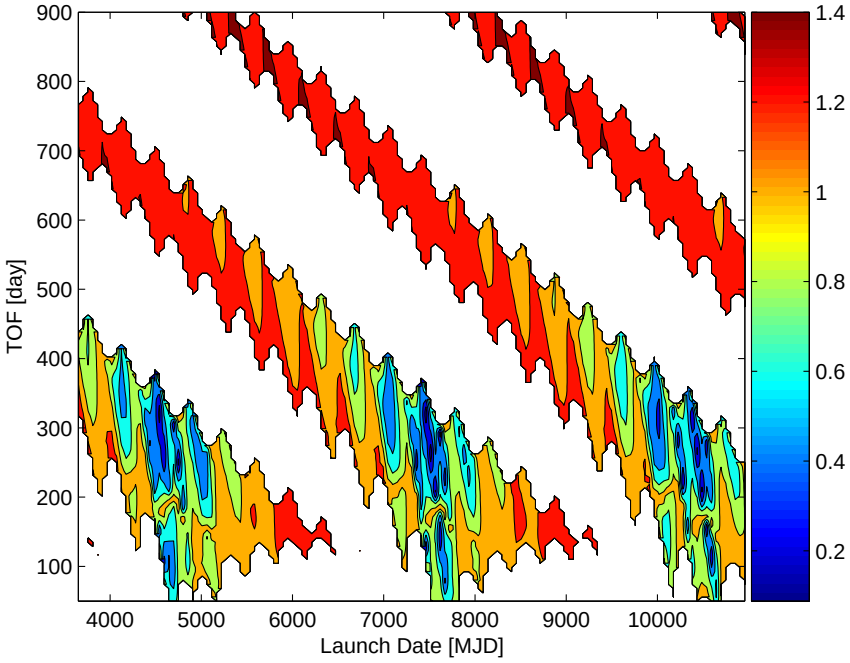


Fig. 3. Objective function J for $k_2 = 0.275$.

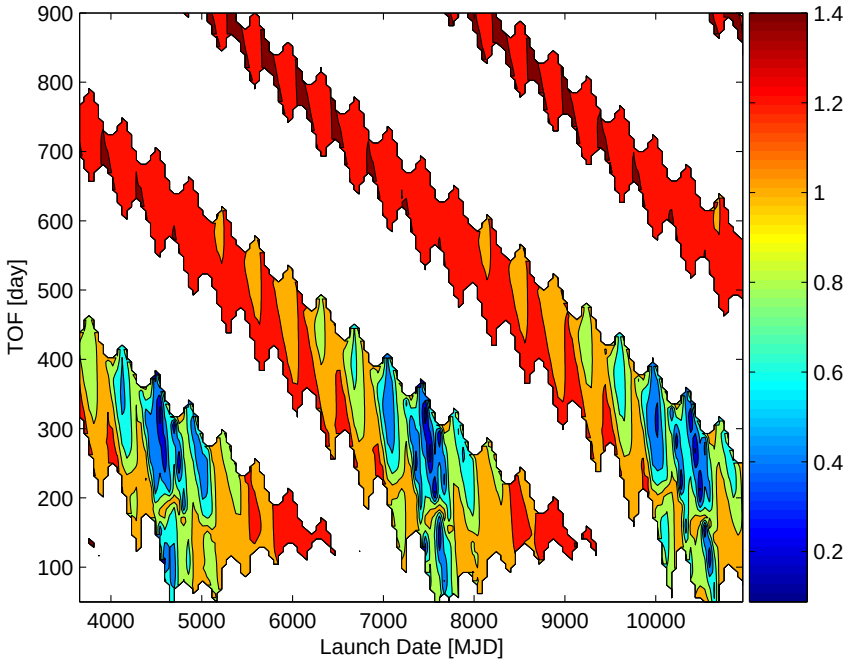


Fig. 4. Objective function J for $k_2 = 0.005$.

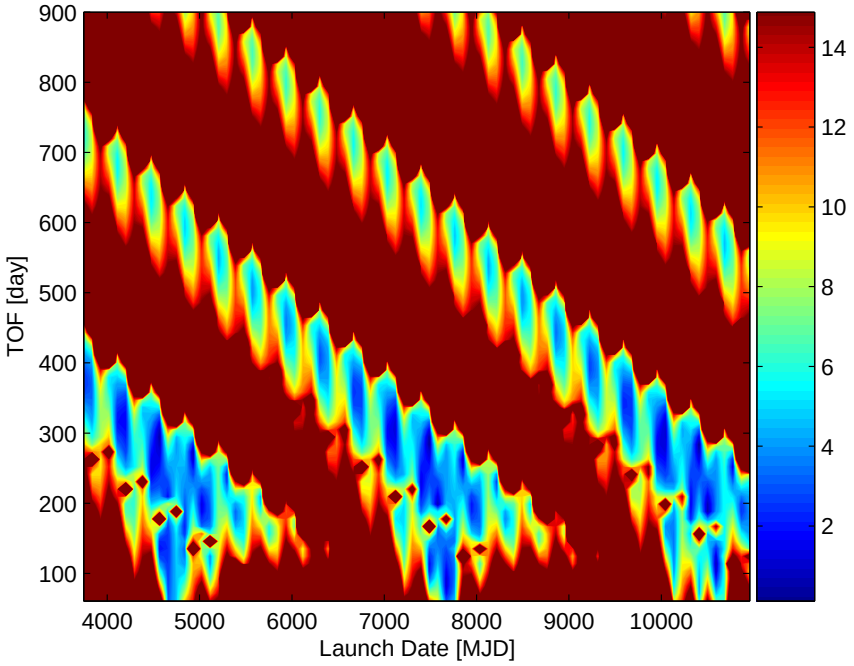


Fig. 5. Total Δv for direct bi-impulsive transfer.

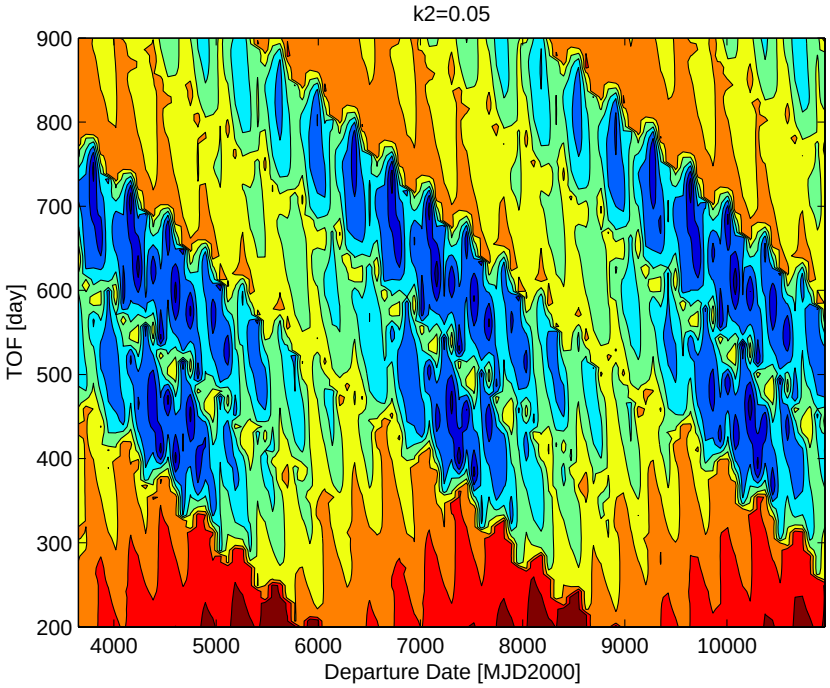


Fig. 6. Objective function J for $k_2 = 0.005$.

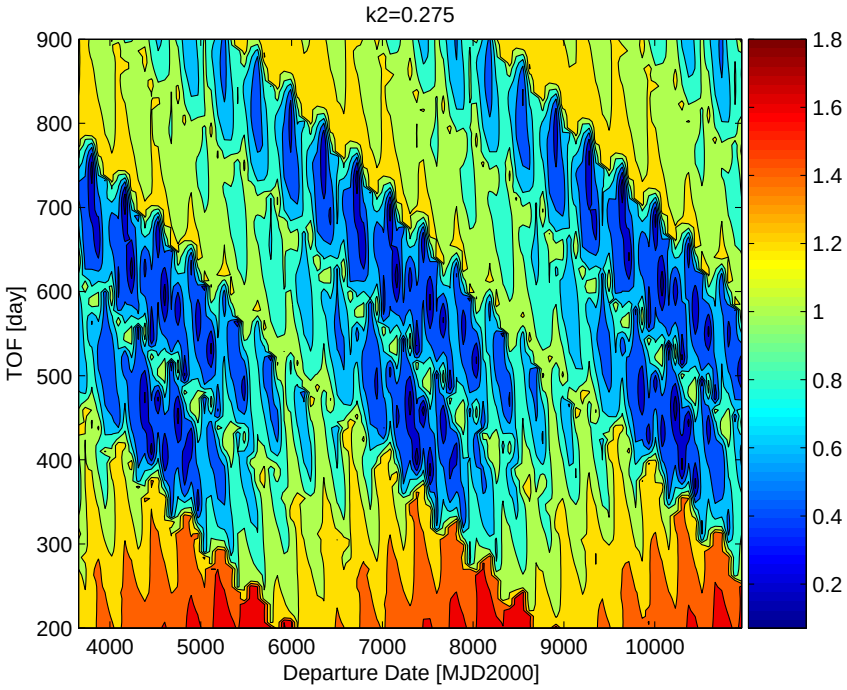


Fig. 7. Objective function$J$ for $k_2 = 0.275$.

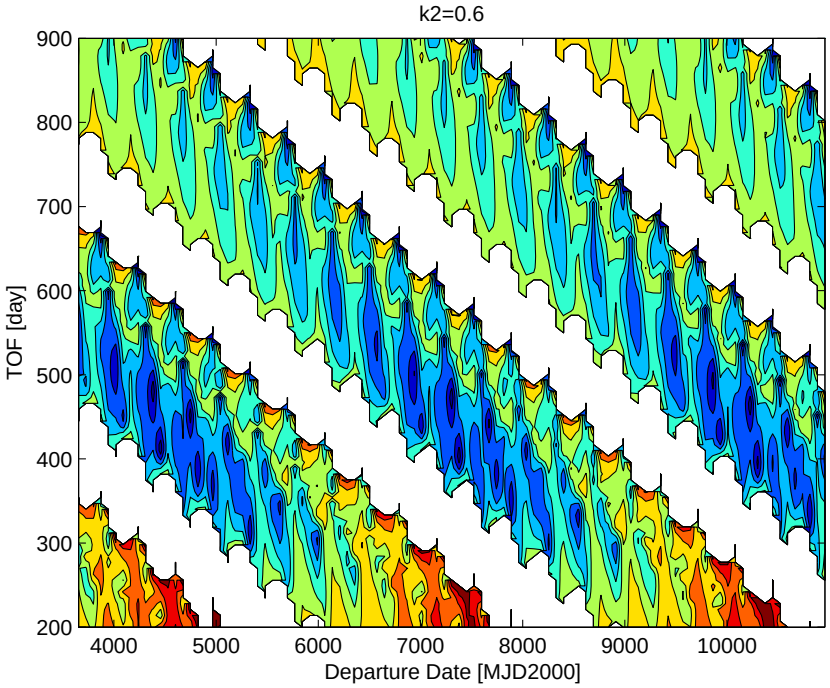


Fig. 8. Objective function J for $k_2 = 0.6$.

in a given transfer time. In order to avoid looping over all possible values of the number of revolutions, a particular heuristic has been implemented: for a given transfer the number of allowable revolutions can not be larger than the ratio between the transfer time and the shortest revolution period between the departure celestial body and the target celestial body. In addition to that, an upper limit on k_2 was derived in order to avoid imaginary solutions. From the condition $1 - k_1 k_2^2 > 0$ we get

$$1 - \frac{k_2^4}{2} \ln^2 \frac{r_1}{r_2} > \cos\left(k_2 \theta_f\right). \tag{3.1}$$

The two curves will cross before

$$k_2 = \frac{2\pi}{\theta_f}. \tag{3.2}$$

Therefore, we start from this point and we look for a solution to the problem

$$1 - \frac{k_2^4}{2} \ln^2 \frac{r_1}{r_2} > \cos\left(k_2 \theta_f\right). \tag{3.3}$$

The upper limit on k_2 reduces the search space but it does not solve the problem completely since it does not identify either the feasible region nor a region in which the cost function is monotonic with respect to k_2. On the other hand, it reduces the size of the search space along the k_2 direction.

3.3 Solution of Lambert's problem with the Exponential Sinusoid

The algorithm for the solution of Lambert's problem with the exponential sinusoid is represented in Fig. 9. Given the initial and final radii r_1 and r_2, the number n of revolutions (or

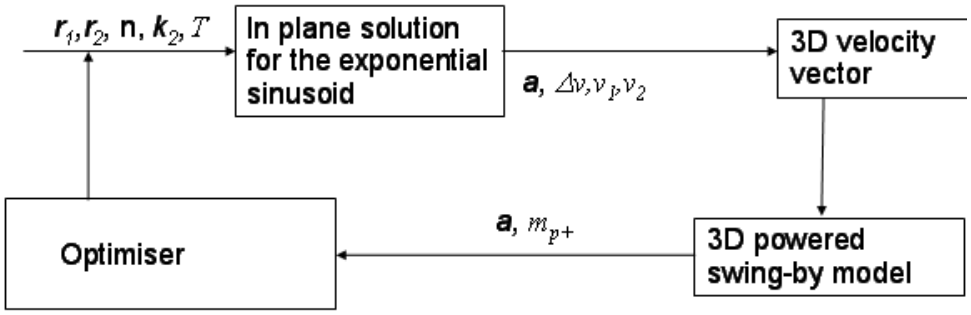


Fig. 9. Algorithm for the solution of the Lambert's problem with the exponential sinusoid

spirals) and the time of flight T, the algorithm computes the acceleration profile a, the total Δv due to the low thrust action and the velocity vectors at the boundaries of the thrust arc v_1 and v_2. The two velocity vectors are in the plane of the orbit and need to be converted into a 3D cartesian reference frame.
After that, the gravity assist manoeuvre is computed with the associated required Δv for the powered swing-by. This Δv and the Δv for the low-thrust arc are finally used to compute the total propellant mass fraction and the objective function.
Part of the code for the solution of the TPBVP with the exponential sinusoid was orignially provided by the ACT and was then modified and integrated with the rest of the software.

3.4 Convergence Analysis

Another analysis of the structural complexity of the search space can be performed looking at the convergence of a simple multistart algorithm. If we call A a generic solution algorithm and p a generic problem we can define the following procedure:

(i) Set the maximum number of function evaluations for A equal to N.
(ii) Apply A to p for n times.
(iii) Compute

$$\phi_{\min}(N) = \min \min f(A(N), p, n)$$
$$\phi_{\max}(N) = \max \min f(A(N), p, n)$$

One expects that if the number of function evaluations N goes to infinity, the two functions converge to the same value, the global minimiser. This can be considered true if the initial set of samples covers densely the whole search space.
Fig. 10 and Fig. 11 show the convergence profile for the bi-impulsive problem, 50 repeated independent runs, Latin hypercube sampling and a local optimisation from each sample. Slightly more than 1000 initial samples are required to have sure convergence to the global minimum.
Fig. 12 and Fig. 13 show the convergence profile for the exponential sinusoid problem, 50 repeated independent runs, Latin Hypercube sampling and a local optimisation from each sample, $k_2 \in [0.05, 0.6], n \in [0, 2]$. After 2000 initial samples there is still a low probability of converging to the minimum.

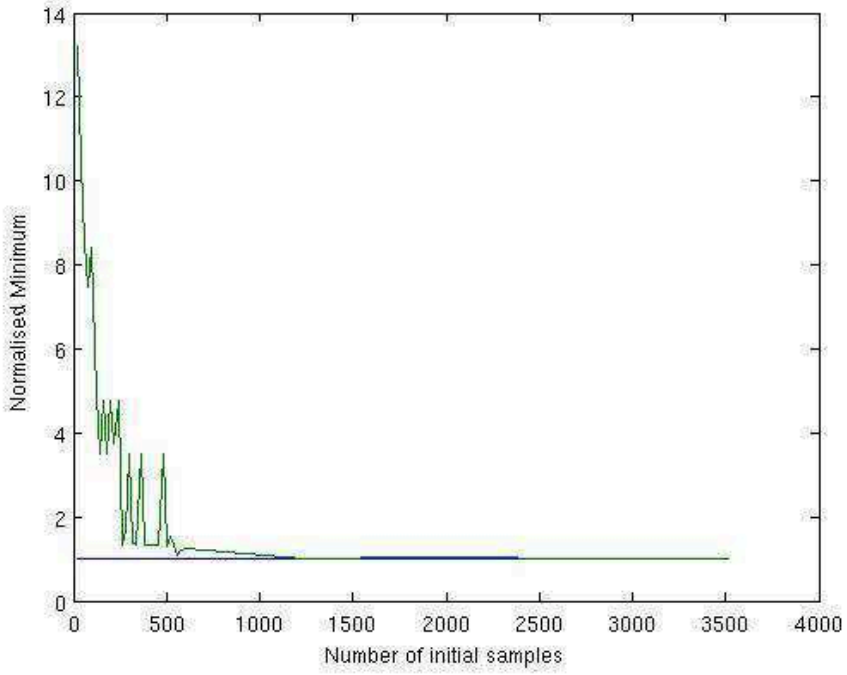


Fig. 10. Convergence of a bi-impulsive direct transfer to Apophis as a function of the number of initial samples.

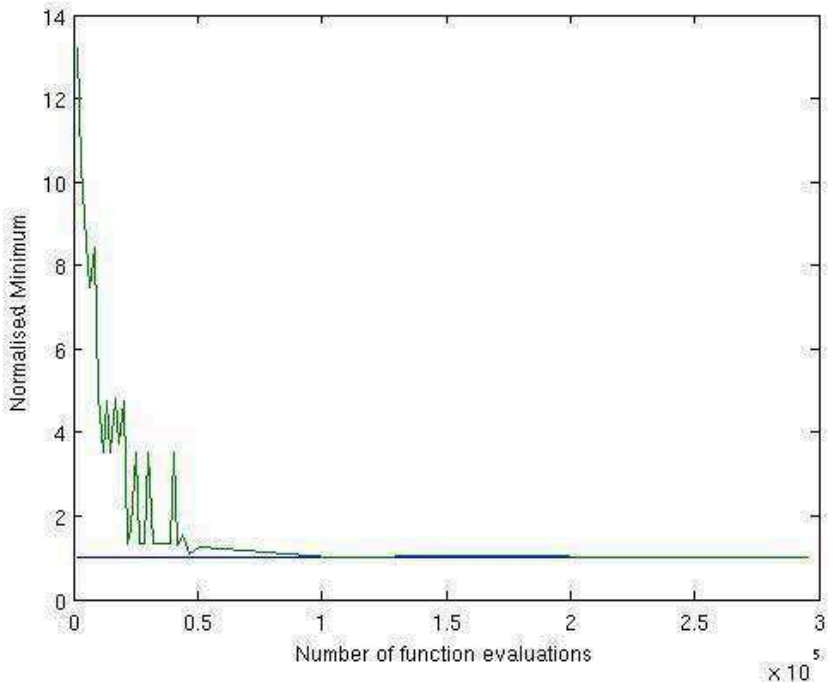


Fig. 11. Convergence of a bi-impulsive direct transfer to Apophis as a function of the total number of function evaluations.

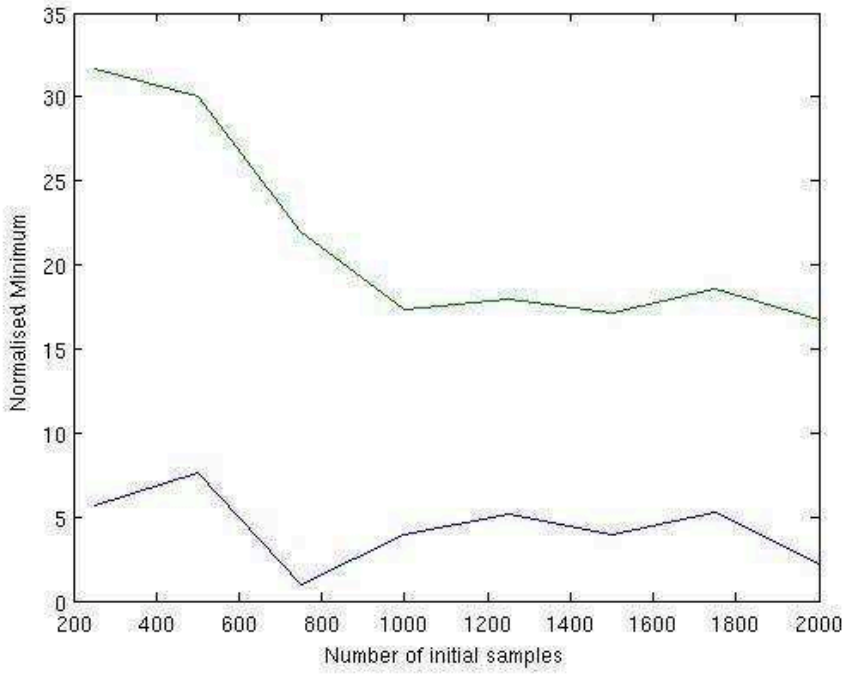


Fig. 12. Convergence of the exposin for a direct transfer to Apophis as a function of the number of initial samples.

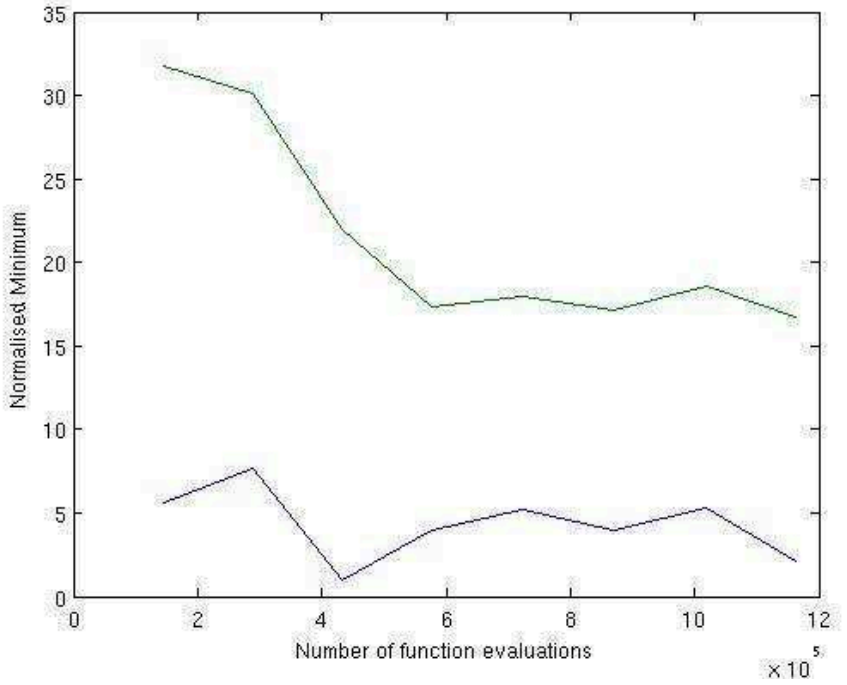


Fig. 13. Convergence of the exposin for a direct transfer to Apophis as a function of the total number of function evaluations.

4 Optimality Analysis

The analysis of the optimality of the solutions has two fundamental aims. The first is to assess how reliable the results are and the second how good they can be as a first guess for a more refined optimisation. In the following we are thus using two different approaches: the analysis of the necessary conditions for optimality derived from Pontryagin's maximum principle and the analysis of the convergence of a direct transcription method initialised with the exponential sinusoid. For the latter case we are using the software DITAN [31].

4.1 Necessary Conditions for Optimality

The optimal control problem underneath the exponential sinusoid can be formulated as follows:

$$\min_F J(F) = \int \frac{1}{2}\frac{F^2}{\dot{\theta}} dt \tag{4.1}$$

subject to

$$\begin{aligned} r' &= \frac{r v_r}{v_\theta} \\ v_r' &= \frac{r}{v_\theta}\left(-\frac{\mu}{r^2} + \frac{v_\theta^2}{r} + F\sin\gamma\right) \\ v_\theta' &= -v_r + \frac{F r \cos\gamma}{v_\theta} \end{aligned} \tag{4.2}$$

where $()'$ denotes the first derivative with respect to the polar angle θ.
The Hamiltonian function is therefore given by:

$$H = \frac{F}{\dot{\theta}} + \lambda_r \frac{r v_r}{v_\theta} + \lambda_{v_r}\frac{r}{v_\theta}\left(-\frac{\mu}{r^2} + \frac{v_\theta^2}{r} + a\sin\gamma\right) + \lambda_{v_\theta}\left(-v_r + \frac{\arccos\gamma}{v_\theta}\right) \tag{4.3}$$

From the Hamiltonian we can derive the optimality conditions:

$$\begin{aligned} o_1 &= \lambda_{v_r}\frac{r}{v_\theta}(-F\cos\gamma) + \lambda_{v_\theta}\left(\frac{F r\sin\gamma}{v_\theta}\right) = 0 \\ o_2 &= \lambda_{v_r}\frac{r}{v_\theta}(\sin\gamma) + \lambda_{v_\theta}\left(\frac{r\cos\gamma}{v_\theta}\right) + \frac{F}{\dot{\theta}} = 0 \end{aligned} \tag{4.4}$$

The first of the two optimality conditions gives us the classical tangent steering law:

$$\tan\gamma = \frac{\lambda_{v_r}}{\lambda_{v_\theta}} \tag{4.5}$$

while the second one gives us the optimal value for the acceleration:

$$F = -\dot{\theta}\left[\lambda_{v_r}\frac{r}{v_\theta}(\sin\gamma) + \lambda_{v_\theta}\left(\frac{r\cos\gamma}{v_\theta}\right)\right]. \tag{4.6}$$

The time history of the adjoint variables can be computed from the necessary conditions:

$$\begin{aligned} \dot{\lambda_{v_r}} &= -\frac{\partial H}{\partial v_r} \\ \dot{\lambda_{v_\theta}} &= -\frac{\partial H}{\partial v_\theta} \\ \dot{\lambda_r} &= -\frac{\partial H}{\partial r} \end{aligned} \tag{4.7}$$

Now if the terminal constraint on the radius is enforced in weak form and there are no terminal constraints on the velocities, for a fixed time problem the final conditions for the adjoint variables are:

$$\begin{aligned} \lambda_{v_r}(\theta = \bar{\theta}) &= \nu_1 \\ \lambda_{v_\theta}(\theta = \bar{\theta}) &= \nu_2 \\ \lambda_r(\theta = \bar{\theta}) &= \nu_3 \end{aligned} \tag{4.8}$$

Since the solution for the exponential sinusoid is providing the states and the controls at any instant of time, the differential equations for the adjoints can be integrated backwards in time. Given the adjoints, the states and the controls, the optimality conditions should be satisfied if the solution is optimal. We can then use the norm of the violation of the optimality conditions as a measure of the optimality of the solution computed with the exponential sinusoid. The violation of the optimality conditions can be defined as:

$$J_2 = \max_{\theta \in [\theta_0, \theta_f]} \int_{\theta_0}^{\theta_f} \left(o_1^2 + o_2^2\right) d\theta \tag{4.9}$$

In order to integrate backwards the adjoint equations, the adjoints ν_1,ν_2 and ν_3 have to be estimated. Given the control law from the exponential sinusoid the following minimisation problem has to be solved:

$$\min J_2(\nu_1, \nu_2, \nu_3) \tag{4.10}$$

The result of this minimisation is the best estimation of the adjoint variables for the imposed control law.

5 LTGASP – Low Thrust Gravity Assist Space Pruning

In this section we propose the LTGASP algorithm which is designed to efficiently detect and prune infeasible parts of the parameter space of a given MLTGA problem. For this, we shortly recall the problem formulation, state the different pruning techniques and finally present the entire algorithm.

5.1 The MLTGA Problem Formulation

In the following we consider the MLTGA optimisation problem which is closely related to (2.8). To be more precise, given an arbitrary but *fixed* sequence $(n_1, \ldots, n_N)$ of numbers of revolutions we consider the following problem:

$$\begin{aligned} \text{minimise:} \quad & J(\tilde{y}) \\ \text{subject to:} \quad & r_p \geq r_{min} \end{aligned} \tag{5.1}$$

where the solution vector $\tilde{y}$ is given by:

$$\tilde{y} = [t_0, T_1, T_2, \ldots, T_N, k_{2,1}, \ldots, k_{2,N}],$$

and where the parameters have the following ranges:

$$\begin{aligned} t_0 &\in \mathcal{I}_0, \\ T_i &\in \mathcal{I}_i,\ i = 1, \ldots, N, \\ k_{2,i} &\in \mathcal{I}_{k_{2,i}},\ i = 1, \ldots, N. \end{aligned}$$

Define $\mathcal{I}$ as the entire search space, i.e.

$$\mathcal{I} := \mathcal{I}_0 \times \ldots \mathcal{I}_N \times \mathcal{I}_{k_{2,1}} \times \ldots \times \mathcal{I}_{k_{2,N}}.$$

Introducing the map (cf. [11])

$$f : x = [t_0, T_1, \ldots, T_N] \to X = [t_0, t_1, \ldots, t_N],$$

defined by the simple component wise relation $t_i = t_0 + \sum_{j=0}^{i} T_j,\ i = 0, \ldots, N$, and setting

$$\mathcal{I}^* := f(\mathcal{I}) \times \mathcal{I}_{k_{2,1}} \times \ldots \times \mathcal{I}_{k_{2,N}}$$

we can reformulate (5.1) as:

$$\begin{aligned} \text{find} \quad & X \in \mathcal{I}^* \\ \text{minimizing} \quad & J(X) \\ \text{subject to} \quad & r_p(X) \geq r_{min}. \end{aligned} \tag{5.2}$$

We first give some notations which are helpful for the statement of the different pruning techniques. Every (feasible) trajectory from planet p_{i-1} to planet p_i in the current setting is determined by the parameters t_{i-1}, t_i, and $k_{2,i}$. Given these three values, denote the resulting trajectory from p_{i-1} to p_i by

$$T(t_{i-1}, t_i, k_{2,i}).$$

Further, denote by $D(\mathcal{I}_i^*)$ and $D(\mathcal{I}_{k_{2,i}})$ the discretisations of $\mathcal{I}_i^*$ and $\mathcal{I}_{k_{2,i}}$. Thus, the entire discretised search space is given by

$$D(\mathcal{I}^*) = D(\mathcal{I}_0^*) \times \ldots \times D(\mathcal{I}_N^*) \times D(\mathcal{I}_{k_{2,1}}) \times \ldots \times D(\mathcal{I}_{k_{2,N}}).$$

Now we are in the position to state the pruning techniques which will be done in the following.

5.2 The Pruning Techniques

In the following we will propose the pruning techniques which are used for the LTGASP algorithm.

Initialisation. Mark all $t_i \in D(\mathcal{I}_i^*),\ i = 0, \ldots, n$, as valid as well as all trajectories

$$\begin{aligned} T(t_{i-1}, t_i, k_{2,i}), \quad & \forall t_{i-1} \in D(\mathcal{I}_{i-1}^*), t_i \in D(\mathcal{I}_i^*), k_{2,i} \in D(\mathcal{I}_{k_{2,i}}), \\ & i = 1, \ldots, N. \end{aligned}$$

***ΔV* constraining.** The maximal allowable ΔV_i is the main pruning criterion of the LTGASP algorithm in phase i. It works on the sampled space $D(I_{i-1}^*) \times D(I_i^*) \times D(I_{k_{2,i}})$ and prunes out all those points corresponding to trajectories having a velocity change larger than a given budget ΔV_i^{max}.
Algorithm 1 describes the ΔV pruning for the transfer from planet p_{i-1} to planet p_i, i. e. for phase i. Denote by $\Delta V_i(T)$ the velocity change required by a given trajectory T.

Algorithm 1 ΔV pruning

for all valid $t_{i-1} \in D(\mathcal{I}^*_{i-1})$ **do**
  for all valid $t_i \in D(\mathcal{I}^*_i)$ **do**
    if $t_i - t_{i-1} \in \mathcal{I}_i$ **then**
      for all $k_{2,i-1} \in D(\mathcal{I}_{k_{2,i-1}})$ **do**
        if $\Delta V_i(T(t_{i-1}, t_i, k_{2,i})) > \Delta V_i^{max}$ **then**
          mark $T(t_{i-1}, t_i, k_{2,i})$ as not valid.
        end if
      end for
    end if
  end for
end for

Departure velocity constraining. This criterion prunes out all trajectories where the departure velocity (and thus the corresponding thrust required by the spacecraft) is larger than a given threshold.

Forward pruning. An application of the ΔV pruning in each phase typically reduces the search space volume of an MLTGA problem significantly. As a consequence many values of the arrival time t_i in phase i become non-feasible departure times in phase $i+1$. To be more precise: if there is no feasible trajectory that arrives at a planet on a given date because they have all been pruned out according to the various criteria introduced, then there will be no departures from that planet on that date. Thus, all the corresponding points will also be pruned.
Algorithm 2 describes the forward pruning from phase i to phase $i+1$ with respect to $t_i \in D(\mathcal{I}^*_i)$.

Algorithm 2 Forward pruning

for all valid $\bar{t}_i \in D(\mathcal{I}^*_i)$ **do**
  If $T(t_{i-1}, \bar{t}_i, k_{2,i})$ is not valid for all $(t_{i-1}, k_{2,i}) \in D(\mathcal{I}^*_i) \times D(\mathcal{I}_{k_{2,i}})$,
  mark $\bar{t}_i$ as not valid as well as all trajectories $T(\bar{t}_i, t_{i+1}, k_{2,i+1})$
end for

Backward constraining. This technique is analogue to the previous one: clearly, if a departure time in phase $i+1$ becomes infeasible because of pruning, also the relative arrival date in phase i has to be pruned out.
Algorithm 3 describes the backward pruning from phase $i+1$ back to phase i with respect to $t_i \in D(\mathcal{I}^*_i)$.

Algorithm 3 Backward pruning

for all valid $\bar{t}_i \in D(\mathcal{I}^*_i)$ **do**
  If $T(\bar{t}_i, t_{i+1}, k_{2,i+1})$ is not valid for all $(t_{i+1}, k_{2,i+1}) \in D(\mathcal{I}^*_{i+1}) \times D(\mathcal{I}_{k_{2,i+1}})$,
  mark $\bar{t}_i$ as not valid as well as all trajectories $T(t_{i-1}, \bar{t}_i, k_{2,i})$
end for

Gravity assist maximum thrust constraint. The gravity assist maximum thrust constraint prunes out the trajectories having a difference between incoming velocities of trajectories in

phase i (denote this velocity by $V^i_{end}(T)$ for a given trajectory T) and outgoing velocities of trajectories in phase $i+1$ (denote by $V^{i+1}_{start}(T)$) during a gravity assist larger than some threshold, A_v. This threshold has to be set separately for each gravity assist. Further, an appropriate tolerance, L_v, based on the Lipschitzian constant of the current phase plot has to be taken into account.
Algorithm 4 describes the gravity assist maximum thrust constraint pruning between phase i and phase $i+1$.

Algorithm 4 Gravity assist maximum thrust constraint pruning

for all valid $\bar{t}_i \in D(\mathcal{I}^*_i)$ **do**
  $v^f_{min} := \min_{t_{i-1},k_{2,i}} V^i_{end}(T(t_{i-1},\bar{t}_i,k_{2,i}))$ ▷ forward
  $v^f_{max} := \max_{t_{i-1},k_{2,i}} V^i_{end}(T(t_{i-1},\bar{t}_i,k_{2,i}))$
  for all valid $t_{i+1} \in D(\mathcal{I}^*_{i+1})$) **do**
    for all valid $k_{2,i+1} \in D(\mathcal{I}^*_{k_{2,i+1}})$) **do**
      if $V^{i+1}_{start}(T(\bar{t}_i,t_{i+1},k_{2,i+1})) \notin [v^f_{min}-A_v-L_v, v^f_{max}+A_v+L_v]$ **then**
        mark $T(\bar{t}_i,t_{i+1},k_{2,i+1})$ as not valid.
      end if
    end for
  end for
  $v^b_{min} := \min_{t_{i+1},k_{2,i+1}} V^{i+1}_{start}(T(\bar{t}_i,t_{i+1},k_{2,i+1}))$ ▷ backward
  $v^b_{max} := \max_{t_{i+1},k_{2,i+1}} V^{i+1}_{start}(T(\bar{t}_i,t_{i+1},k_{2,i+1}))$
  for all valid $t_{i-1} \in D(\mathcal{I}^*_{i-1})$) **do**
    for all valid $k_{2,i} \in D(\mathcal{I}^*_{k_{2,i}})$) **do**
      if $V^i_{end}(T(t_{i-1},\bar{t}_i,k_{2,i})) \notin [v^b_{min}-A_v-L_v, v^b_{max}+A_v+L_v]$ **then**
        mark $T(t_{i-1},\bar{t}_i,k_{2,i})$ as not valid.
      end if
    end for
  end for
end for

Gravity assist angular constraint. The gravity assist angular constraint prunes infeasible swingbys from the search space on the basis of them being associated with a hyperbolic periapse under the minimum safe distance for the given gravity assist body.
This is determined over every arrival date $\bar{t}_i \in D(\mathcal{I}^*_i)$ as follows: for all incoming trajectories $T(t_{i-1},\bar{t}_i,k_{2,i})$ and all outgoing trajectories $T(\bar{t}_i,t_{i+1}k_{2,i+1})$ check if the corresponding swingby is valid. In this case mark both incoming and outgoing trajectory as valid. Finally (i.e. after going through all arrival dates), all trajectories not marked as valid by this procedure will be pruned out.
Algorithm 5 describes the gravity assist angular constraint pruning between phase i and phase $i+1$.

Breaking manoeuvre constraint. As well as the departure velocity constraint, it is logical to add a constraint on the maximum breaking manoeuvre that a spacecraft can perform and prune out trajectories with an exceedingly high fuel demand.

Algorithm 5 Gravity assist angular constraint pruning

```
for all $\bar{t}_i \in D(I_i^*)$ do
    for all valid incoming trajectories $T(t_{i-1}, \bar{t}_i, k_{2,i})$ do
        for all valid outgoing trajectories $T(\bar{t}_i, t_{i+1}, k_{2,i+1})$ do
            if the swingby for $T(t_{i-1}, \bar{t}_i, k_{2,i})$ and $T(\bar{t}_i, t_{i+1}, k_{2,i+1})$ is valid then
                mark $T(t_{i-1}, \bar{t}_i, k_{2,i})$ as valid
                mark $T(\bar{t}_i, t_{i+1}, k_{2,i+1})$ as valid
            end if
        end for
    end for
end for
Invalidate all trajectories not marked as valid
```

5.3 The LTGASP Algorithm

Having stated the different pruning techniques we are now able to state the complete pruning algorithm. Given an MLTGA problem (5.2), the LTGASP algorithm for the search space reduction reads as follows:

(0) perform the initialisation process.
(1) perform the ΔV pruning, departure velocity pruning as well as the forward pruning (one "phase shift") for phase 1.
(2) for $l = 2, \ldots, n-1$
  (a) perform the ΔV pruning for phase l.
  (b1) perform the backward pruning from phase l downto phase 1.
  (b2) perform the forward pruning from phase 1 up to phase $l+1$.
  (c) perform the gravity assist pruning for phases $l-1$ and l.
  (d) perform the angular constraint pruning for phases $l-1$ and l.
(3) perform all the pruning steps described in step (2) plus the breaking manoeuvre constraint for phase n.

Remark. This is just one possible way to combine the different pruning techmiques. Note that the steps 2(a), 2(b) and 2(c) can be interchanged, and that the outcome of the resulting pruning algorithm depends on this choice. However, since the angular constraint pruning is the most time consuming technique (see subsequent sections), it is logical to apply this technique at last in each phase. Further, it is also possible e.g. to apply the backward/forward pruning after each crucial pruning criterion (such as the angular constraint pruning). This technique is typically quite effective and has – in general – a running time which is almost negligible compared to the other pruning criteria.

6 Time and Space Complexity for the LTGASP Algorithm

This section determines the time and space complexity of the LTGASP algorithm. It will be shown that LTGASP scales quadratically in space and quintic in time with respect the number of gravity assist manoeuvres considered. For simplicity, the following analysis assumes that the initial launch window and all phase times are the same.

6.1 Space Complexity

Consider a launch window, a mission phase time, and the range of the $k_2's$ discretised into l bins. Thus, for the first phase l^3 Lambert problems have to be sampled. Since the number of

possible times the planet may be arrived at in phase i, $i = 1, \ldots, n$, can be assumed to be $(i+1)l$ (see [11]) the i-th phase will require an amount of $(i+1)l \cdot l \cdot l = (i+1)l^3$ Lambert function evaluations (given by the discretisations of the departure times, the time of flight and k_2). This gives the series

$$l^3 + 2l^3 + \ldots + nl^3 = l^3 \frac{n(1+n)}{2} = O(n^2).$$

Therefore, the amount of space required for n phases is only of order $O(n^2)$, rather than $O(k^{2n+1})$ for full grid sampling. Similarly, the space complexity with respect to the resolution l is $O(l^3)$.

6.2 Time Complexity

The memory space requirement is directly proportional to the maximum number of Lambert problems that must be solved, and hence the time complexity of the sampling portion of the LTGASP algorithm must also be of the order $O(n^2)$.
For the further time complexity analysis we make the following assumptions (see above):

$$\begin{aligned} &|D(\mathcal{I}_i^*)| = (i+1)l, \quad i = 0, \ldots, n, \\ &|D(I_{k_{2,i}})| = l, \quad i = 1, \ldots, n. \end{aligned} \tag{6.3}$$

***ΔV* constraint complexity.** The i-th step requires on the order of i^2l^3 operations since by (6.3) it follows that $|D(\mathcal{I}_{i-1}^*)| \cdot |D(\mathcal{I}_i^*)| \cdot |D(I_{k_{2,i}})| \approx i^2l^3$. Thus, the time complexity applying the ΔV constraint is $O(n^3)$ with respect to the dimensionality and $O(l^3)$ with respect to the resolution.

Forward and backward pruning complexity. The forward pruning requires on the order of i^2l^3 flops for one "phase shift" (i.e. the pruning of the deptarture times for phase $i+1$ by analysing the data of phase i). Since for every phase i there are i such phase shifts (i.e. after the ΔV pruning of phase i and after the backward pruning), for this pruning criterion an amount of

$$\sum_{j=1}^{n} \sum_{i=1}^{j} i^2l^3 = l^3 \sum_{j=1}^{n} \frac{j(j+1)(2j+1)}{6}$$

flops is required. Thus, the complexity of the forward pruning is $O(n^4)$ with respect to the dimensionality and $O(l^3)$ with respect to the resolution. Analogously, the complexity of the backward pruning is of the same order.

Gravity assist thrust constraint complexity. The i-th step requires of the order of $2i^2l^3$ operations. Thus, the time complexity applying the gravity assist thrust constraint is $O(n^3)$ with respect to the dimensionality and $O(l^3)$ with respect to the resolution.

Gravity assist angular constraint complexity. The i-th step requires of the order of $2i^3l^5$ operations. Thus, the time complexity applying the gravity assist angular constraint is $O(n^4)$ with respect to the dimensionality and $O(l^5)$ with respect to the resolution.

Overall time complexity. The overall complexity, taken from the most complex part of the algorithm (the gravity assist angular constraint), is quintic with respect to the resolution and quartic with respect to the dimensionality.

7 Testing LTGASP

In this section we show the results for four different example missions. The computations were done on an Intel Xeon 3.2 Ghz processor.

7.1 Sequence EVE

First we consider the sequence Earth – Venus – Earth. As critical parameters we have chosen the ones shown in Table 1. The MATLAB commands for the execution of the pruning read as follows:

```
P=ltgasp([3,2,3],[4745 5840],[100 200;300 400],[80 100 120], ...
         [0.01 2; 0.01 2],[20 20], [5 5],[30 30], 6750, 1);

pplot3 (P);
```

Figure 14 shows the resulting 'candidate set', i.e. the set of points which were not pruned out by the LTGASP algorithm[2], and Table 2 shows the corresponding running times.

Table 1. Parameter settings for the pruning of the EVE sequence.

number of celestial bodies	:	3
sequence	:	Earth – Venus – Earth
launch window	:	$[4745, 5840]$ (days after $01.01.2000$)
time of flight phase 1	:	$[100, 200]$ (days)
time of flight phase 2	:	$[300, 400]$ (days)
$\|D(\mathcal{I}_0^*)\|$	:	80
$\|D(\mathcal{I}_1^*)\|$	:	100
$\|D(\mathcal{I}_2^*)\|$	:	120
range of $k_{2,1}$	:	$[0.01, 2]$
range of $k_{2,2}$	:	$[0.01, 2]$
$\|D(I_{k_{2,1}})\|$	:	20
$\|D(I_{k_{2,2}})\|$	:	20
ΔV_1^{max}	:	$5\ (m/s)$
ΔV_2^{max}	:	$5\ (m/s)$
max. departure velocity	:	$30\ (m/s)$
max. terminal velocity	:	$30\ (m/s)$
rp_{min} Venus	:	$6750\ (m)$

7.2 Sequence EVEJ

Next we consider the extended sequence Earth – Venus – Earth – Jupiter. Choosing analogous pruning parameters as in the previous example the commands for executing the LTGASP algorithm and displaying the candidate set read as follows:

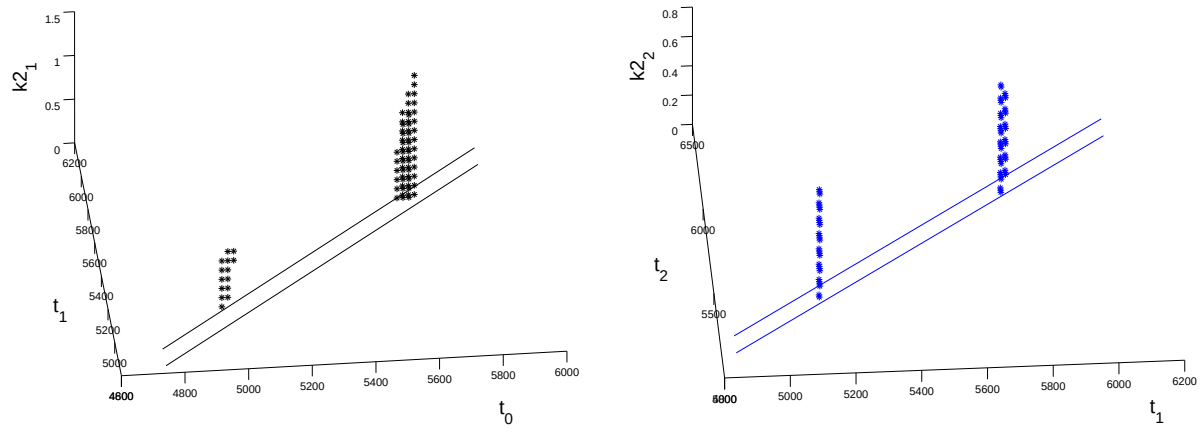
```
P=ltgasp([3,2,3,5], [4745 5840], [100 200;300 400;500 1500], ...
         [80, 100, 120, 140], [0.01 2; 0.01 2;0.01 2], ...
         [20, 20, 20], [5, 5], [100, 100, 100], [6750, 6750], 1);

pplot3 (P);
```

Figure 15 and Table 4 show the resulting candidate set and the corresponding running time respectively.

Table 2. Number of function calls and running time for sequence EVE

Function calls	:	
exposin-calls	:	14,379
gravity assists	:	42,467
Running times	:	
ΔV pruning 1. phase	:	429.37 sec.
ΔV pruning 2. phase	:	335.93 sec.
backward/forward pruning	:	0.02 sec.
max. thrust pruning	:	6.45 sec.
ang. constr. pruning	:	3475.93 sec.
2nd backward/forward pruning	:	0.02 sec.
final backw./forw. pruning	:	3.87 sec.
total running time	:	4251.59 sec.



(a) Phase 1 (b) Phase 2

Fig. 14. Numerical result of the LTGASP algorithm on the EVE sequence.

Then more realistic values for the departure and arrival velocities were used and the launch date was extended (see Table 5). Table 6 shows the resulting running time.

7.3 Sequence EVEJE

As the third example we make another extension of the previous one and consider the sequence EVEJE. Again similar settings for the pruning parameters were used. An application of the LTGASP algorithm leads to a candidate set which is displayed in Figure 16. The corresponding running times are shown in Table 7.

7.4 Sequence EVM

Finally we consider the sequence Earth – Venus – Mercury. Using the pruning parameters shown in Table 8 the following MATLAB commands have to be used to display the candidate set (see Figure 17):

[2] This set is obviously in general a strict superset of the feasible set, but hopefully a good approximation of it.

Table 3. Parameter settings for the pruning of the EVEJ sequence: first test.

number of celestial bodies	:	4
sequence	:	Earth – Venus – Earth – Jupiter
launch window	:	$[4745, 5840]$ (days after $01.01.2000$)
time of flight phase 1	:	$[100, 200]$ (days)
time of flight phase 2	:	$[300, 400]$ (days)
time of flight phase 3	:	$[1000, 2000]$ (days)
$\|D(\mathcal{I}_0^*)\|$	:	80
$\|D(\mathcal{I}_1^*)\|$	:	100
$\|D(\mathcal{I}_2^*)\|$	:	120
$\|D(\mathcal{I}_3^*)\|$	:	100
range of $k_{2,1}$	:	$[0.01, 2]$
range of $k_{2,2}$	:	$[0.01, 2]$
range of $k_{2,3}$	:	$[0.01, 2]$
$\|D(I_{k_{2,1}})\|$	:	20
$\|D(I_{k_{2,2}})\|$	:	20
$\|D(I_{k_{2,3}})\|$	:	20
ΔV_1^{max}	:	$5\ (m/s)$
ΔV_2^{max}	:	$5\ (m/s)$
ΔV_3^{max}	:	$5\ (m/s)$
max. departure velocity	:	$30\ (m/s)$
max. terminal velocity	:	$30\ (m/s)$
rp_{min} Venus	:	$6750\ (m)$
rp_{min} Earth	:	$6750\ (m)$

Table 4. Number of function calls and running time for sequence EVEJ: first test.

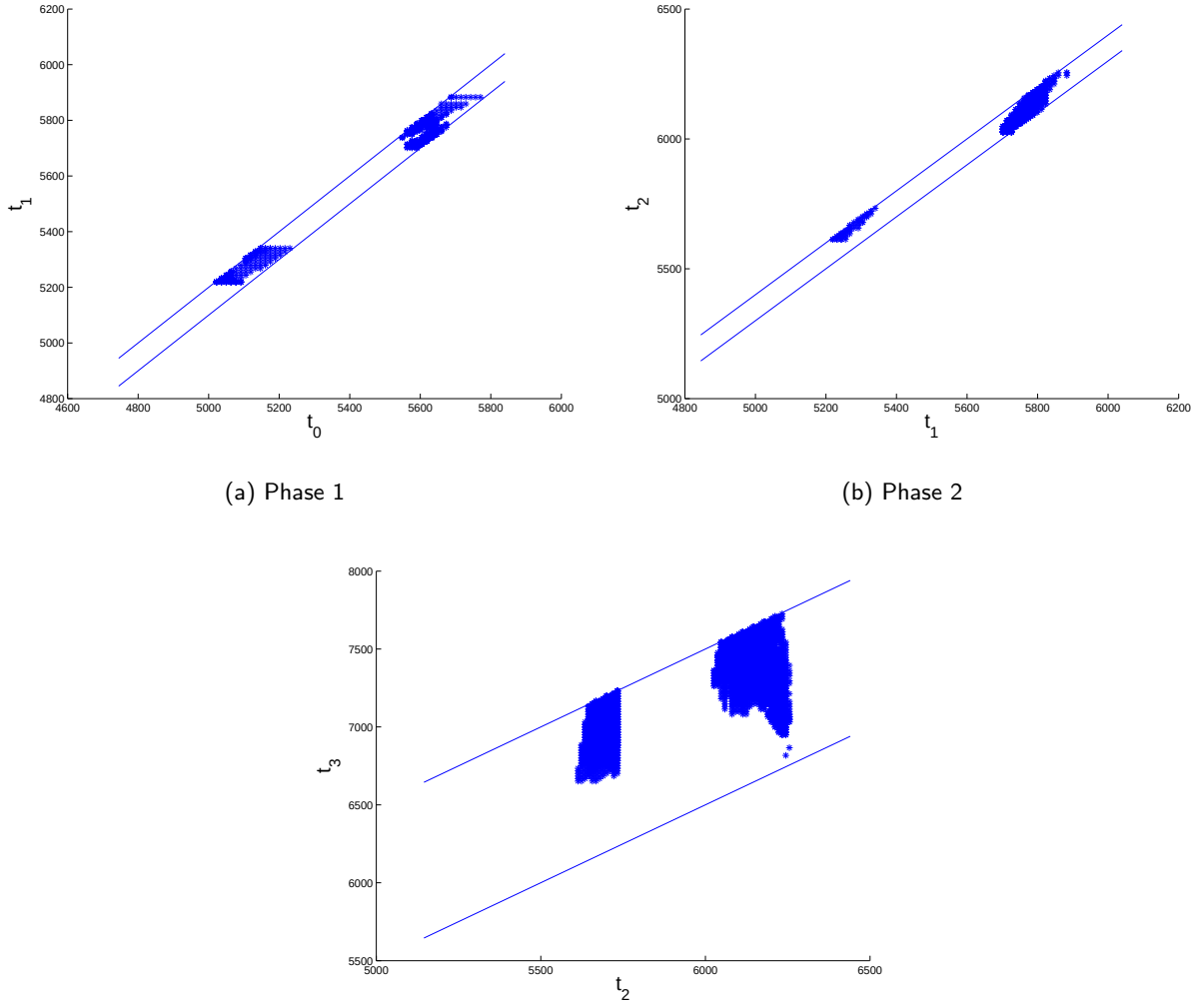
Function calls	:	
exposin-calls	:	24317
gravity assists	:	4543
Running times	:	
ΔV pruning 1. phase	:	431.56 sec.
ΔV pruning 2. phase	:	340.24 sec.
backward/forward pruning	:	0.03 sec.
max. thrust pruning	:	5.06 sec.
ang. constr. pruning	:	3475.93 sec.
backward/forward pruning	:	3.87 sec.
ΔV pruning 3. phase	:	1710.13 sec.
backward/forward pruning	:	0.05 sec.
max. thrust pruning	:	9.14 sec.
ang. constr. pruning	:	2373.77 sec.
backward/forward pruning	:	0.06 sec.
final backw./forw. pruning	:	5.60 sec.
total running time	:	8355.70 sec.

Table 5. Parameter settings for the pruning of the EVEJ sequence: second test.

number of celestial bodies	:	4
sequence	:	Earth – Venus – Earth – Jupiter
launch window	:	$[3650, 7300]$ (days after $01.01.2000$)
time of flight phase 1	:	$[100, 200]$ (days)
time of flight phase 2	:	$[300, 400]$ (days)
time of flight phase 3	:	$[1000, 2000]$ (days)
$\|D(\mathcal{I}_0^*)\|$	:	1040
$\|D(\mathcal{I}_1^*)\|$	:	100
$\|D(\mathcal{I}_2^*)\|$	:	100
$\|D(\mathcal{I}_3^*)\|$	:	100
range of $k_{2,1}$	:	$[0.01, 2]$
range of $k_{2,2}$	:	$[0.01, 2]$
range of $k_{2,3}$	:	$[0.01, 2]$
$\|D(I_{k_{2,1}})\|$	:	20
$\|D(I_{k_{2,2}})\|$	:	20
$\|D(I_{k_{2,3}})\|$	:	20
ΔV_1^{max}	:	$15\ (m/s)$
ΔV_2^{max}	:	$15\ (m/s)$
ΔV_3^{max}	:	$15\ (m/s)$
max. departure velocity	:	$5\ (m/s)$
max. terminal velocity	:	$30\ (m/s)$
rp_{min} Venus	:	$6750\ (m)$
rp_{min} Earth	:	$6750\ (m)$

Table 6. Number of function calls and running time for sequence EVEJ: second test.

Function calls	:	
exposin-calls	:	43497
gravity assists	:	55964
Running times	:	
ΔV pruning 1. phase	:	2361.43 sec.
ΔV pruning 2. phase	:	120.35 sec.
backward/forward pruning	:	0.3 sec.
max. thrust pruning	:	35.57 sec.
ang. constr. pruning	:	35127.82 sec.
backward/forward pruning	:	0.35 sec.
ΔV pruning 3. phase	:	473.83 sec.
backward/forward pruning	:	0.34 sec.
max. thrust pruning	:	4.15 sec.
ang. constr. pruning	:	730.98 sec.
backward/forward pruning	:	0.31 sec.
final backw./forw. pruning	:	3.675 sec.
total running time	:	38857.20 sec.



(a) Phase 1

(b) Phase 2

(c) Phase 3

Fig. 15. Numerical result of the LTGASP algorithm on the EVEJ sequence.

```
P = ltgasp([3,2,1],[700 1300],[100 800;500 1500],[100,120,140], ...
           [.01 2;.01 2], [20 20], [5,5],[50,100], 750, 1);

pplot3 (P);
```

8 Optimization

In this section we address the optimization problem. In particular, we propose how the pruning techniques can be integrated efficiently into the optimization process. Hereby we consider both the 'classical' scalar optimization problem described above as well as a related multi-objective optimization problem.

8.1 Scalar Optimization

In the following we consider the MLTGA problem (5.2). In the course of the study the optimization process was performed by following the subsequent steps:

Table 7. Number of function calls and running time for sequence EVEJE

Function calls	:	
exposin-calls	:	14,379
gravity assists	:	42,467
Running times	:	
ΔV pruning 1. phase	:	423.82 sec.
ΔV pruning 2. phase	:	338.61 sec.
backward/forward pruning	:	0.04 sec.
max. thrust pruning	:	6.62 sec.
ang. constr. pruning	:	3582.48 sec.
backward/forward pruning	:	0.04 sec.
ΔV pruning 3. phase	:	1688.62 sec.
backward/forward pruning	:	0.08 sec.
max. thrust pruning	:	9.09 sec.
ang. constr. pruning	:	2364.60 sec.
backward/forward pruning	:	0.08 sec.
ΔV pruning 4. phase	:	2575.04 sec.
backward/forward pruning	:	0.10 sec.
max. thrust pruning	:	19.40 sec.
ang. constr. pruning	:	7471.00 sec.
backward/forward pruning	:	0.10 sec.
final backw./forw. pruning	:	7.65
total running time	:	18487.37 sec.

Table 8. Parameter settings for the pruning of the EVM sequence.

number of celestial bodies	:	3
sequence	:	Earth – Venus – Mercury
launch window	:	$[700, 1300]$ (days after $01.01.2000$)
time of flight phase 1	:	$[100, 800]$ (days)
time of flight phase 2	:	$[500, 1500]$ (days)
$\lvert D(\mathcal{I}_0^*)\rvert$	:	100
$\lvert D(\mathcal{I}_1^*)\rvert$	:	120
$\lvert D(\mathcal{I}_2^*)\rvert$	:	140
range of $k_{2,1}$	:	$[0.01, 2]$
range of $k_{2,2}$	:	$[0.01, 2]$
$\lvert D(I_{k_{2,1}})\rvert$	:	20
$\lvert D(I_{k_{2,2}})\rvert$	:	20
ΔV_1^{max}	:	5 (m/s)
ΔV_2^{max}	:	5 (m/s)
max. departure velocity	:	50 (m/s)
max. terminal velocity	:	100 (m/s)
rp_{min} Venus	:	6750 (m)

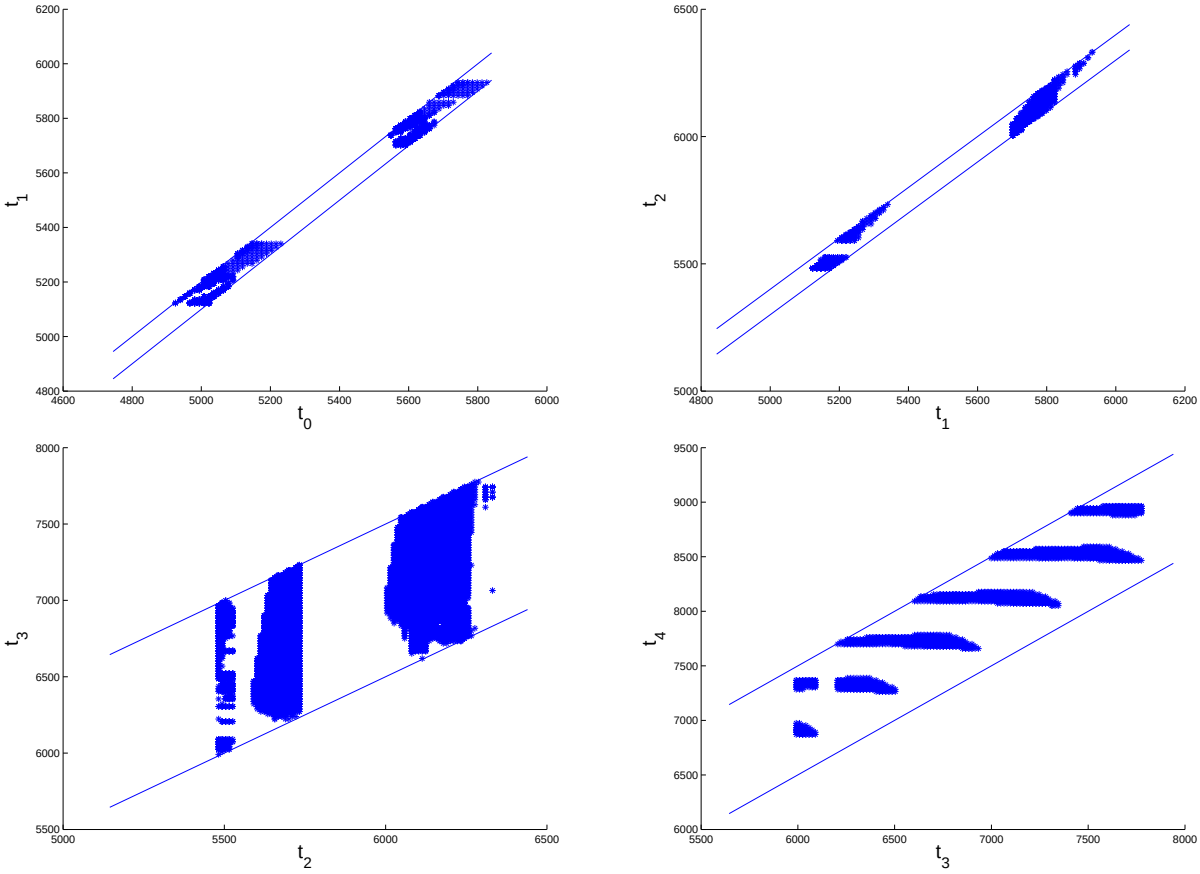


Fig. 16. Numerical result of the LTGASP algorithm on the EVEJE sequence.

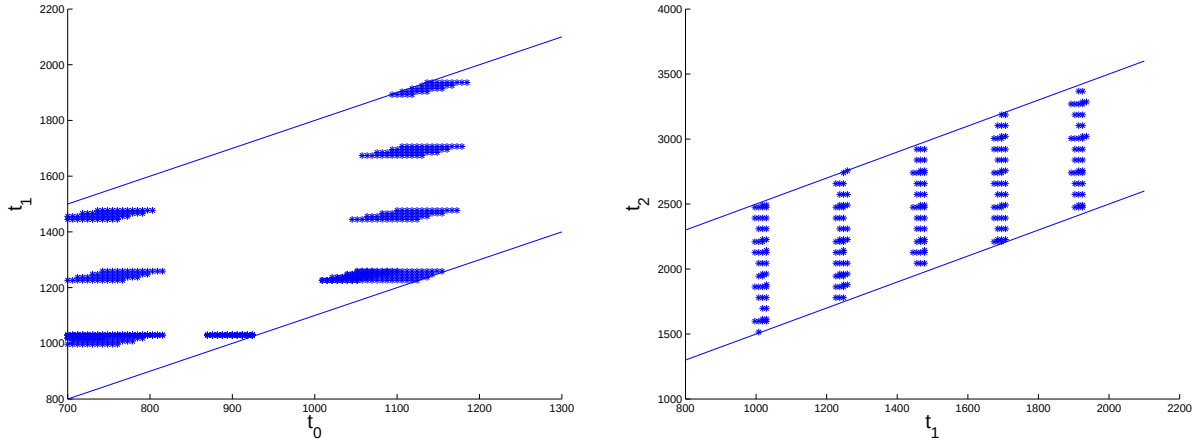
Table 9. Number of function calls and running time for sequence EVM.

Function calls	:	
exposin-calls	:	156,864
gravity assists	:	36,534
Running times	:	
ΔV pruning 1. phase	:	3940.79 sec.
ΔV pruning 2. phase	:	3884.91 sec.
backward/forward pruning	:	0.03 sec.
max. thrust pruning	:	15.15 sec.
ang. constr. pruning	:	9613.19 sec.
2nd backward/forward pruning	:	0.03 sec.
final backw./forw. pruning	:	5.93 sec.
total running time	:	17460.03 sec.

(1) Perform the LTGASP algorithm. Denote the resulting candidate set by $\mathcal{C}$.
(2) Construct a *box collection* $\mathcal{R}$ starting from $\mathcal{C}$, i.e. every element $B \in \mathcal{R}$ is of the form

$$B = B(c, r) = \{x \in \mathbb{R}^N \, : \, |x_i - c_i| \leq r_i \text{ for } i = 1, \ldots, N\}.$$

This box collection should certainly cover $\mathcal{C}$. Further, in order to speed up the computation, it is desired that (a) the volume of $\mathcal{R}$ is small, and (b) the number of boxes is small. The elements of $\mathcal{R}$ can for instance be detected by looking at the connected components of the logical 3-dimensional matrices A_i which correspond to the i-th phase: set $a_{jkl} = 0$ if $T(t_j, t_k, k_{2_l})$ is not valid for the underlying sequence, where t_j is the j-th element of



(a) Phase 1 (b) Phase 2

Fig. 17. Numerical result of the LTGASP algorithm on the EVM sequence.

$|D(\mathcal{I}^*_{i-1})|$ (t_k and k_{2_l} analogous); else set $a_{jkl} = 1$. The boxes can e.g. be selected by taking the minimal and maximal coordinate values of each connected component (see Figure 18 for an example). The amount of boxes[3] can be reduced by merging neighboring connected components.

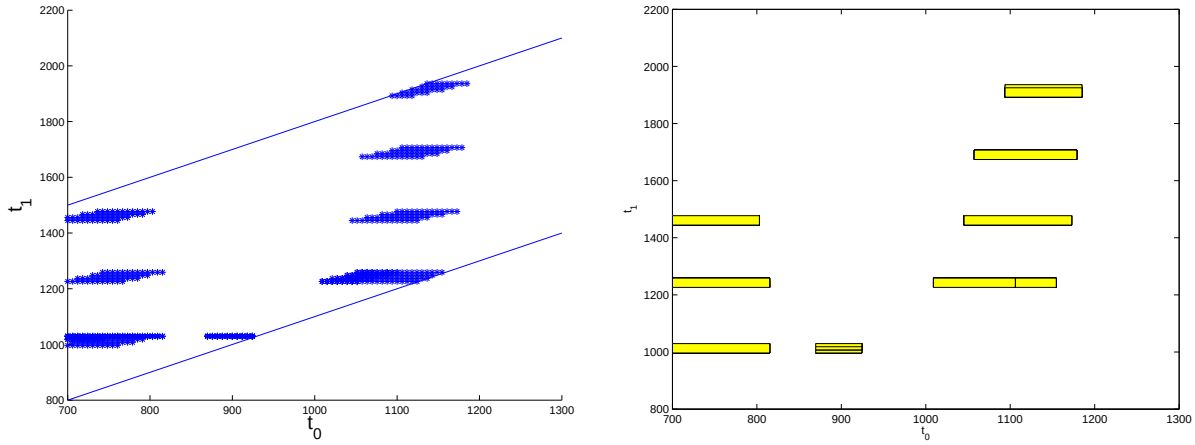
(3) Perform a preliminary analysis on *all* boxes of $\mathcal{R}$. This can e.g. be done by running a (separate) global scalar optimization algorithm on every box. Since these boxes were constructed after an application of the pruning algorithm, the optimization problem within each box can be (roughly) regarded to be unconstrained. Thus, in principle every global optimization algorithm which can handle box constraints can be applied.
In this study we have performed the preliminary analysis by executing a Differential Evolution algorithm ([25]) of short running time.

(4) Continue the search on the boxes which are 'promising' after the preliminary analysis in Step (3) (e.g., by considering the best found function value in the box, and/or the volume of the box, etc.).

Remark. Since the number of boxes can get large, it would be desirable to be able to perform *one* run of a global optimization algorithm on the entire box collection (as it was done in the multi-objective case, see below). Unfortunately, there was not enough time within this project to develop an efficient algorithm adapted to this particular case. However, this could be an interesting task for further studies.
As an example we consider the sequence EVEJ using the constraints and pruning parameters described in Section 7.2. An application of Step (2) leads to an amount of 11 boxes (using `Q_label3.m`). The preliminary analysis (`prune_and_eval.m`) detected the following interval as the best one:

```
interval 1:
t_0: [5548.924051, 5770.696203]
t_1: [5702.020202, 5858.939394]
t_2: [6026.470588, 6255.000000]
```

[3] Using the method described above up to several hundreds of boxes have been obtained for some settings.



(a) Candidate set (b) Box collection

Fig. 18. Relaxation of the candidate set (left) by using boxes (right), which are easier to handle for most optimization algorithms.

```
t_3: [6949.352518, 7725.359712]
volume:  6171502693.164243
bestval: 0.920646
bestmem: 5624.269722 5755.441374 6103.010469 7136.829305
```

In a longer DE run on this interval the best value could be improved significantly (from $J_1 = 0.920646$ to $J_2 = 0.72786$). The best parameters found are:

$$\begin{aligned}(t_0, t_1, t_2, t_3) &= (5571.18, 5765.25, 6118.45, 7426.74)\\ (k_{2,1}, k_{2,2}, k_{2,3}) &= (0.5780, 0.8084, 0.6312).\end{aligned} \tag{8.4}$$

In an analogous run where different bounds for the pruning were used the following parameters were detected

$$\begin{aligned}(\tilde{t}_0, \tilde{t}_1, \tilde{t}_2, \tilde{t}_3) &= (3875.53, 4088.72, 4495.98)\\ (\tilde{k}_{2_1}, \tilde{k}_{2_2}, \tilde{k}_{2,3}) &= (0.8, 0.8, 0.5),\end{aligned} \tag{8.5}$$

corresponding to a mass fraction of $\tilde{J} = 0.6845$.

8.2 Multi-Objective Optimization

In addition to the (scalar) MLTGA problem defined in Section 5.1 we have considered the related *multi-objective optimization problem* (MOP, see e.g. [6] or [3] for an introduction), since their solution sets (so-called *Pareto sets*) typically offer much more information to the decision maker (DM) of the design process. To be more precise, in the current case it could be interesting to take in addition to the mass fraction also the transfer time into into account since it seems to be more and more crucial for current space missions to have cheap *and* fast transfers (this problem has also been considered by [4] and [12]). Thus, the following MOP has to be considered:

$$\begin{aligned} \text{find:} \quad & X \in \mathcal{I}^* \\ \text{minimizing:} \quad & \begin{cases} J(X) \\ t_N - t_0 \end{cases} \\ \text{subject to:} \quad & r_p(X) \geq r_{min}. \end{aligned} \tag{8.6}$$

In order to obtain the Pareto set of a given multi-objective MLTGA problem we have followed the first two steps of the optimization process described for the scalar case. That is, we have constructed box collections on the base of the candidate sets. In the next step, we have used the subdivision technique proposed in [8] for the computation of the Pareto sets since for this algorithm the boxes can directly be used (in fact, the algorithm generates a sequence of box collections which converge toward the Pareto set under certain assumptions).
Figures 19 and 20 show the Pareto fronts for two sequences. In both cases we have taken the same setting as in Section 7. The results indicate that it definitely worth looking at the corresponding multi-objective optimization problem since the fronts cover a significant range according to both objective values.

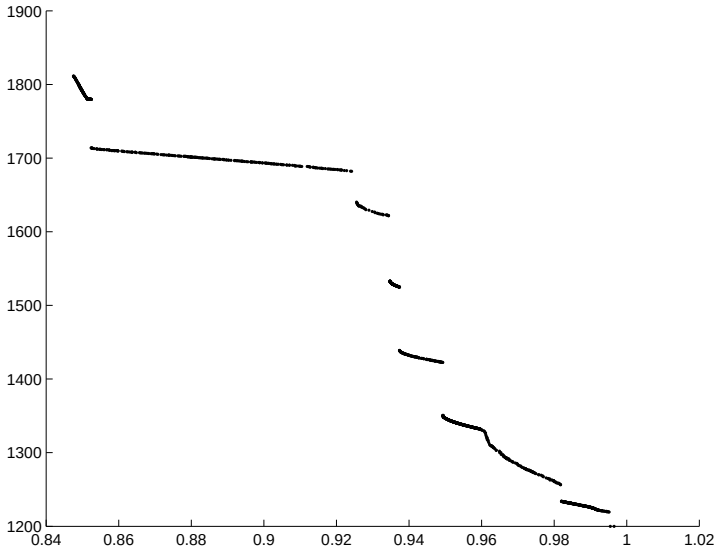


Fig. 19. Pareto front for the EVM sequence.

9 An Alternative Approach: The Pseudo-Equinoctial Shaping

A low-thrust arc can be modeled as a variation of a purely conic arc. For a conic arc the position vector of a spacecraft can be written by means of non-singular equinoctial elements $\alpha^T = [p, f, g, h, k, L]$:

$$\mathbf{r} = \begin{bmatrix} \frac{r}{1+h^2+k^2}(\cos L + (h^2 - k^2)\cos L + 2hk \sin L) \\ \frac{r}{1+h^2+k^2}(\sin L + (h^2 - k^2)\sin L + 2hk \cos L) \\ \frac{2r}{1+h^2+k^2}(h \sin L - k \cos L) \end{bmatrix} \tag{9.7}$$

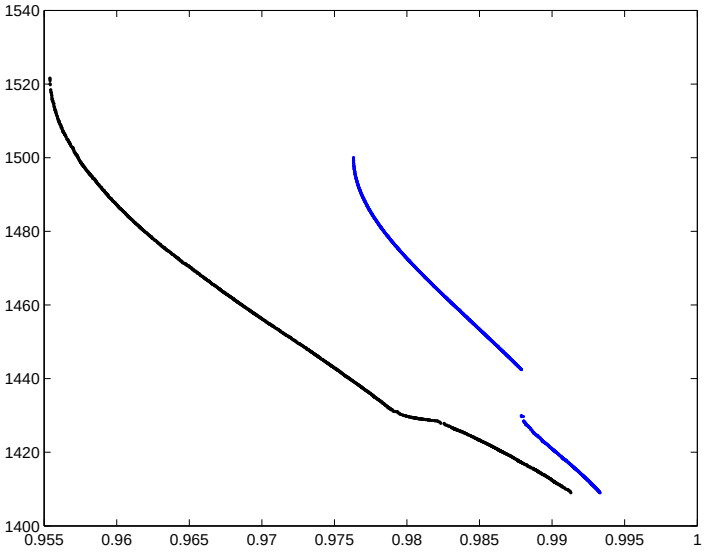


Fig. 20. Pareto front for the EVEJ sequence.

The expression in (9.7) was derived assuming that the orbital parameters were osculating. On the other hand we can see (9.7) as a particular parameterization of the position vector in terms of an arbitrary set of functions of L that we call pseudo-equinoctial elements $\widetilde{\alpha}^T = [\widetilde{p}, \widetilde{f}, \widetilde{g}, \widetilde{h}, \widetilde{k}, L]$. These elements are here called pseudo-equinoctial because they do not satisfy exactly Gauss' planetary equations unless the thrust is zero. Every pseudo-element can be expressed as a function of L in the following form:

$$\alpha = \alpha_0 + \delta\alpha. \tag{9.8}$$

From the definition of r, velocity v and accelerations a can be computed by differentiation:

$$\begin{aligned} \mathbf{v} &= \frac{d\mathbf{r}}{dt} = \frac{d\mathbf{r}}{dL}\frac{dL}{dt}, \\ \mathbf{a} &= \frac{d\mathbf{v}}{dt} = \frac{d\mathbf{v}}{dL}\frac{dL}{dt}, \\ \frac{d\mathbf{r}}{dL} &= \sum_{i=1}^{5} \frac{\partial \mathbf{r}}{\partial \alpha_i}\frac{\partial \alpha_i}{\partial L} + \frac{\partial \mathbf{r}}{\partial L}. \end{aligned} \tag{9.9}$$

In order to obtain the set of pseudo-elements that satisfies exactly the conditions at the boundaries, the following nonlinear programming problem must be solved:

$$\mathbf{r}(\alpha(L_0), L_0) = \mathbf{r}_0, \tag{9.10}$$

$$\mathbf{r}(\alpha(L_f), L_f) = \mathbf{r}_f. \tag{9.11}$$

On the other hand, for small values of the acceleration it is sufficient to solve the easier linear problem

$$\alpha(L_0) = \alpha_0. \tag{9.12}$$

Of course the thrust profile, though constrainable, is a direct consequence of the shape and must be considered only as a first guess useful for a further, more refined optimization. The

optimality of the thrust profile depends on the imposed shape. In previous studies [18] two shapes were proposed, an exponential one:

$$\alpha_i = \alpha_{0,i} + \alpha_{1,i} e^{\lambda_i (L - L_0)}, \quad i = 1, \ldots, 5, \tag{9.13}$$

with $\lambda_2 = \lambda_3$ and $\lambda_4 = \lambda_5$, and a linear trigonometric one:

$$\alpha_i = \alpha_{0,i} + \alpha_{1,i}(L - L_0) + \lambda_i \sin(L - L_0 + \psi), \quad i = 1, \ldots, 5, \tag{9.14}$$

again with $\lambda_2 = \lambda_3$ and $\lambda_4 = \lambda_5$ and ψ a constant value.
For each set of pseudo-elements a different trajectory can be generated, connecting two points in state space. The necessary controls to achieve the imposed shape of the trajectory and the resulting mass expenditure can be obtained with the following formulas:

$$\mathbf{a}_c = \ddot{\mathbf{r}}(\alpha(L)) + \frac{\mathbf{r}(\alpha(L))}{r(\alpha(L))^3} \tag{9.15}$$

$$m_p = 1 - \exp\left(- \int_{L_0}^{L_f} \frac{|\mathbf{a}_c|}{I_{sp} g_0} \frac{dt}{dL} dL \right). \tag{9.16}$$

The total time of flight can be computed from the time rate of the true longitude L:

$$\Delta t = \int_{L_0}^{L_f} \frac{dt}{dL} dL. \tag{9.17}$$

The actual time of flight Δt must be equal to the required time of flight, therefore the following physical constraints must hold:

$$\Delta t - TOF = 0. \tag{9.18}$$

Table 10 reports a comparison between the solutions found with the pseudo-equinoctial elements and the optimal control solution for a direct Earth-Mars transfer. The optimal solution was computed by the Mission Analysis section at ESOC for the CDF study of the mission Exomars. As can be noticed the pseudo-elements are estimating both the time of flight and the propellant consumption very well.

Table 10. EM: Comparison with Optimal Control Solution

Data	EM-1	EM-2	EM-3	CDF
Launch	06/03/07	03/03/07	14/03/07	18/0.3/07
TOF (day)	745	764	751	873
Prop. Mass Ratio	0.168	0.167	0.162	0.156

9.1 Solution of the Lambert's problem with the Pseudo-Equinoctial Elements

The proposed shaping technique needs to define the value of three shaping parameters for each thrust arc. Each shaping parameter controls a group of orbital parameters. In order to solve the Lambert problem and get a physical trajectory connecting two points in space we have to find the right set of shaping parameters that strictly satisfy (9.18). The algorithm for the solution of the Lambert's problem with the pseudo-elements is represented in Fig. 21.

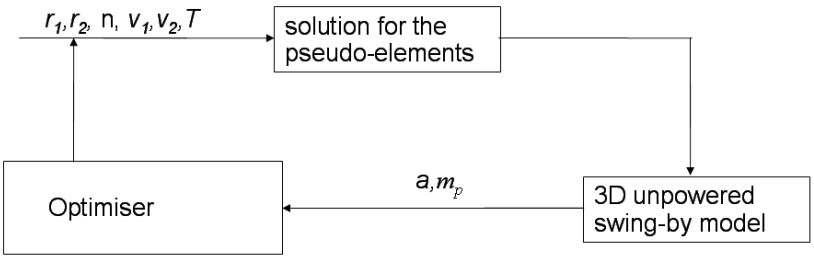


Fig. 21. Algorithm for the solution of the Lambert's problem with the pseudo-elements

Unlike the exponential sinusoid, the pseudo-equinoctial model requires the definition of the values of the velocity vectors at the boundaries. The required input is therefore made of the two position vectors at the boundaries $\mathbf{r}_1$ and $\mathbf{r}_2$, the two velocity vectors at the boundaries $\mathbf{v}_1$ and $\mathbf{v}_2$, the time of flight and the number of revolutions. In the first part of the loop the time constraint is solved and the propellant mass is minimised by changing the values of the shaping parameters. The output is the thrust profile and the propellant mass fraction. Since the value of the velocity vectors at the boundaries can be specified, there is no need for a powered swing-by model for the gravity manoeuvre. We use a linked conic model that satisfies exactly the physical conditions for a gravity assist manoeuvre, for more details about this gravity assist model, the interested read can refer to [18].
As stated above, the first part of the loop requires the solution of the time constraint. Fig. 22 shows the Δt as a function of λ_1 and λ_2, the two shaping parameters controlling the semimajor axis and the eccentricity of the orbit, for a fixed value of λ_3. The surface is smooth and monotonic and its intersection with the plane at the desired TOF can be seen in Fig. 23. All the points along the red line in Fig. 23 represent feasible solutions for the Lambert's problem. One of them minimises the propellant mass consumption, therefore the problem becomes to find the values of λ_1 and λ_2 that minimise the propellant consumption and satisfy (9.18).
Table 11 reports the computational cost for the solution of the minimum mass Lambert's problem for different times of flight. The problem was solved in Matlab with the function $fmincon$. The computational time T_{conv} represent the time required to converge to an optimal solution on an Athlon 64 under Linux.
The minimum mass Lambert problem was solved following two approaches. Approach one, in the table, solves boundary constraints (9.11) while approach two solves boundary constraints (9.12). In the latter case the mass consumption is underestimated since the conditions on the velocity at the boundaries are not perfectly matched, on the other hand the computational time is even up to 50 lower than in the former case.

Table 11. EM: Computational Cost for the Lambert Problem

TOF (Day)	Approach 1	Approach 2
600	m_p=257.4 kg T_{conv}=12.62s	m_p=209.72 kg T_{conv}=9.38s
700	m_p=234.4 kg T_{conv}=10.53s	m_p=195.31 kg T_{conv}=8.76s
800	m_p=229.4 kg T_{conv}=7.41s	m_p=200.25 kg T_{conv}=3.85s
900	m_p=219.4 kg T_{conv}=7.25s	m_p=192.13 kg T_{conv}=7.72s
1000	m_p=210.3 kg T_{conv}=7.82s	m_p=192.51 kg T_{conv}=9.71s

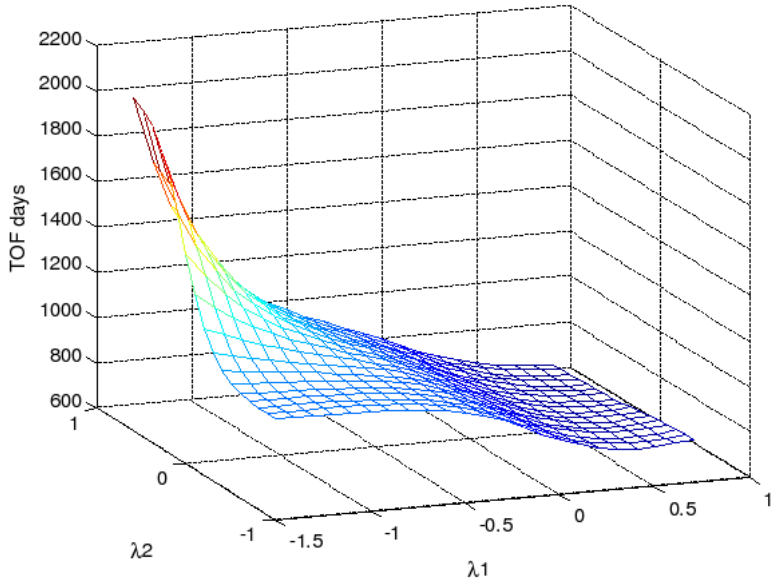


Fig. 22. Δt as a function of λ_1 and λ_2.

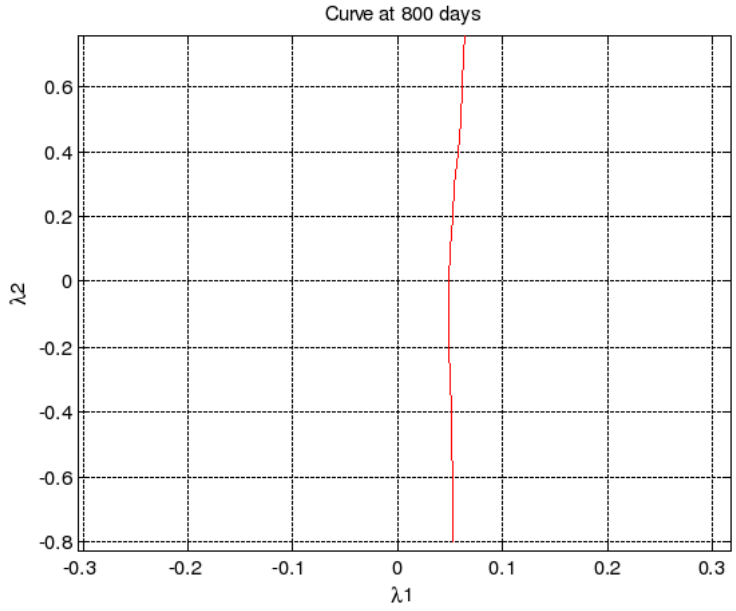


Fig. 23. $\Delta t = TOF$ as a function of λ_1 and λ_2.

—

10 An Alternative Approach: Search Through EPIC

The effectiveness of the pruning technique has been compared to other search methods that are not making any use of problem dependent information. We compare the output of the combination of the pruning technique and Differential Evolution (DE) to a multistart application of DE. Finally we apply a hybrid approach implemented in the software code EPIC [14, 18, 32].

EPIC combines a stochastic search with a deterministic domain decomposition technique. The decomposition of the solution space is driven by the outcome of the stochastic search.
The stochastic part of the algorithm deploys a number of virtual agents [13] in the search space. Each agent applies a number of behavioral rules in order to take actions and share information with the other agents.
The result of the exploration by the agents is passed to the domain decomposition part of the algorithm that takes decisions on how to decompose the domain and where (in which portion of the domain) to deploy other agents for a further exploration. All the three approaches were applied to the EVEJ transfer in the following way:

(1) Pure DE: strategy 6, 40 independent runs, 20000 function evaluations each for a total of 800000 function evaluations, 20 individuals;
(2) DE+Pruning: 20000 function evaluations on each of the feasible sets, 20813 exposin evaluations for pruning the search space, 7 generated intervals, 20 individuals;
(3) EPIC: 20000 function evaluations for each domain exploration, Maximum 10 explored domains, 20 agents.

The result of these three tests can be seen in Table 12. Though the combined Pruning+DE makes use of only a fraction of the function evaluations in comparison to the multistart DE the solution is comparable. Furthermore the interesting thing is that the interval launch dates and encounter times identified by the pruning contains all the solutions reported in the table. EPIC performs well though with a total number of function evaluations higher than the combined Pruning+DE.

Table 12. EVEJ: Comparison with Alternative Search Methods

Pruning+DE	Multistart DE	EPIC
0.6681	0.6374	0.6349

11 Optimality Analysis

In this section we test the optimality of the solutions that can be obtained with the two different shape-based methods. We compare the exponential sinusoid approach to the pseudo-element approach. The solutions from the two approaches are used both to initialize a direct transcription method and to initialize an indirect transcription method.

According to optimal control theory, the solution of the optimal control problem equivalent to the exponential sinusoid yields the optimality conditions for the thrust steering angle and the thrust modulus represented in Fig. 24 and Fig. 25, respectively. Fig. 24 represents the optimal steering law and the steering law provided by the exponential sinusoid for a direct Earth to Mars transfer. As can be clearly seen the control law provided by the exponential sinusoid is far from being optimal.
Fig. 25 represents the thrust modulus. Even in this case the optimal solution is far from the one provided by the exponential sinusoid. The two thrust laws are so different that it was not possible to initialize the solution of the necessary equations for optimality. The first guess for the optimal control laws in the previous figures was actually computed with a direct collocation method.

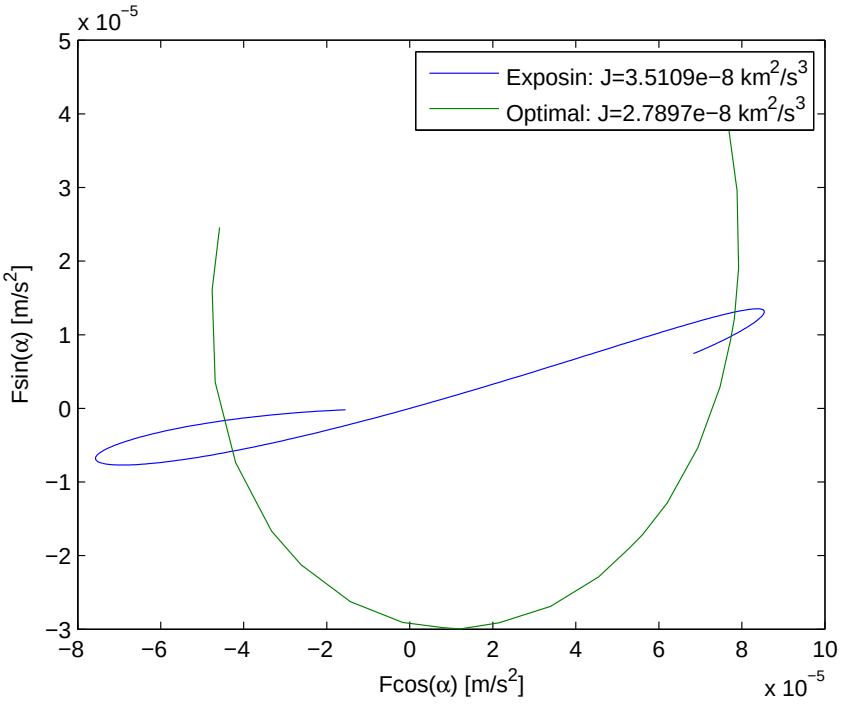


Fig. 24. Optimality Condition on the steering angle for the Exponential Sinusoid.

Unlike what can be obtained with the exponential sinusoid with the pseudo-elements the estimation of the control laws could be used to solve the necessary conditions for optimality. The comparison between the optimal solution and the shaped one can be seen in Figs. 26, 27 and 28 for the three components of the thrust vector. As can be seen, in this case the estimated law and the optimal one are very similar, in particular they have the same frequency and similar amplitude.
Therefore, though the pseudo-elements require a higher computational cost in comparison to the exponential sinusoid they provide a suboptimal solution that is closer to the local optimal one. This is true for the specific case of the solution of the minimum control problem but it holds also true for minimum mass problems. Of course in the latter case the optimal solution is generally, if no singular arcs occur, a switched control law, i.e. the engine is switched on and off, and this type of structure can not be reproduced either with the pseudo-elements nor with the exponential sinusoid.

Table 13. EM: Comparison with DITAN

Data	Pseudo Elements	DITAN
TOF (day)	800	800
Prop. Mass Ratio	0.1704	0.1681

If the solution of the exponential sinusoid was not useful as first guess to solve the necessary conditions for optimality, it was, on the other hand, effective in initializing a direct collocation method. Here we used the software code DITAN to optimize a multiple gravity assist low-thrust transfer to Jupiter. The GA sequence is Venus-Earth and the departure date, transfer time and planetary encounter dates have been estimated through the pruning approach com-

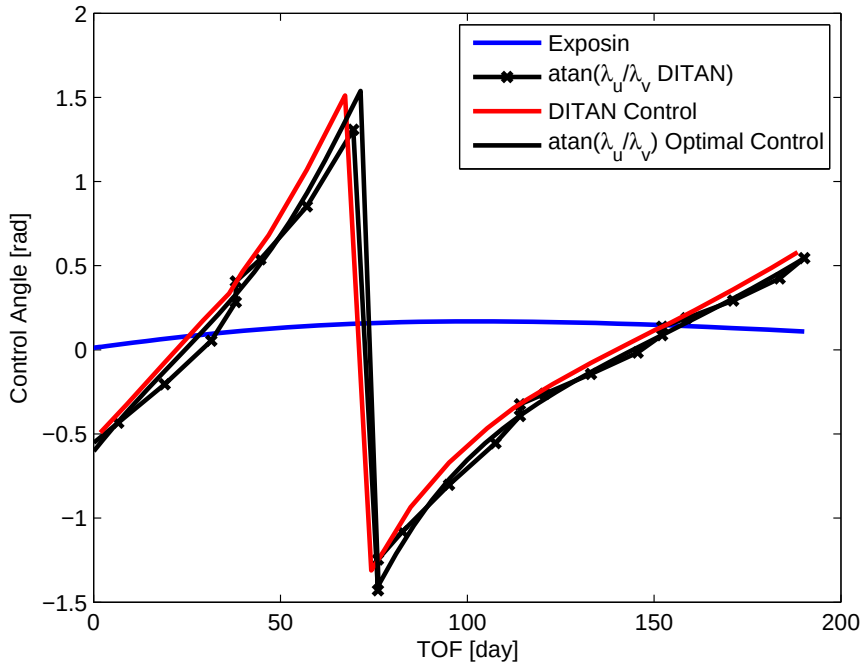


Fig. 25. Optimality Condition on the thrust for the Exponential Sinusoid.

bined with differential evolution. The combination of pruning and differential evolution (called pruning+DE in the tables) provided a number of solutions, two of the best were used to initialize DITAN and the result can be read in Table 14. The objective function is the integral of the square of the modulus of the control and the departure dates has been constrained to remain in a small interval containing the value provided by the exponential sinusoid while the transfer times have been left free. As can be read the optimized solution is similar to the first guess though the gain in propellant (last row) is quite significant. The huge difference of the propellant mass fraction in the last column with respect to the third one is due to the following reason. The Δv manoeuvres for the powered-swingbys in the first solution computed with the exponential sinusoid are not zero nor close to zero. The optimized solution, therefore, has to compensate for the mismatched entry and exit conditions for each swing-by. On the other hand the second first guess has the Δv manoeuvres at every powered-swingbys almost zero. The optimized solution, therefore, simply improve the thrust arcs. The gain in propellant confirm that the exponential sinusoid is far from providing an optimal control law for low-thrust arcs.

12 Conclusions

This study proposed an incremental branch and prune algorithm that can improve the MLTGA problem by reducing the search space in polynomial time when the low-thrust arcs are modeled through the exponential sinusoid and a powered-swingby model is used for the gravity assist manoeuvres.

The first part of the study showed how the structure of the solution space for low-thrust transfers is only partially different from the structure of a bi-impulsive transfer. The main effect due to low-thrust arcs is to blur the basins of attraction of the minima that already

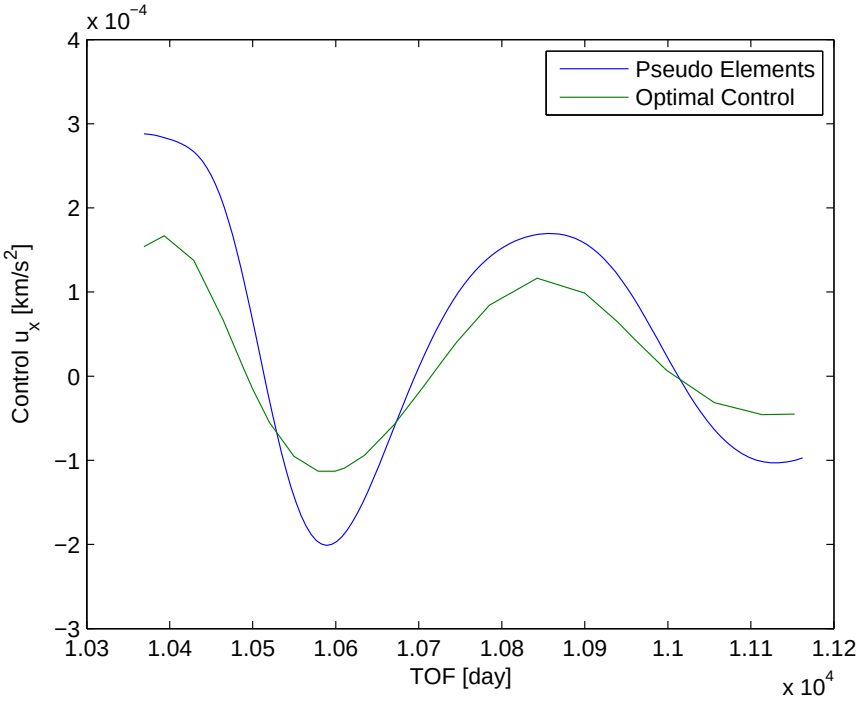


Fig. 26. Optimality Condition on u_x for the Pseudo Elements.

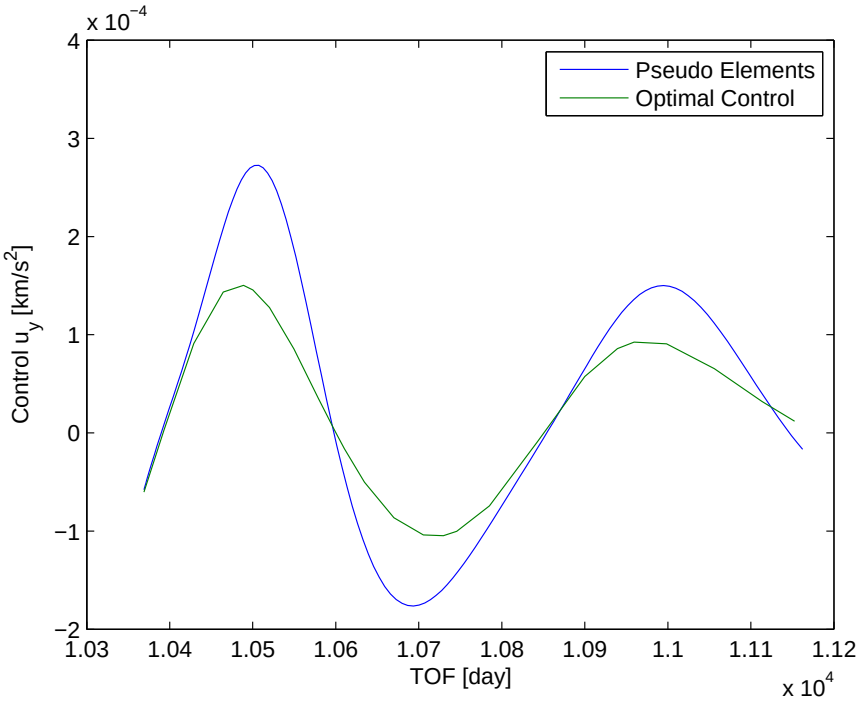


Fig. 27. Optimality Condition on u_y for the Pseudo Elements.

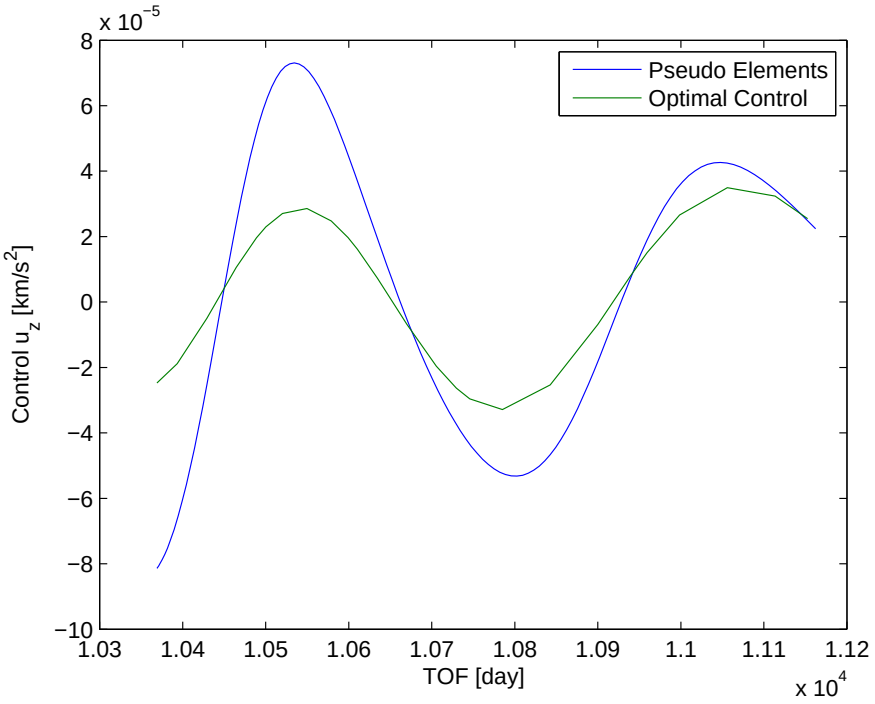


Fig. 28. Optimality Condition on u_z for the Pseudo Elements.

Table 14. EVEJ: Minimum Control Problem

Parameter	Pruning+DE 1 400mN	Pruning+DE 2 300mN	DITAN 1 Min Control	DITAN 2 Min Control
Launch Date (MM-DD-YY)	June 27, 2010	June 27, 2018	June 27, 2010	July 01, 2018
Departure Velocity (km/s)	0.54119	1.118	0.54	1.1
Venus TOF (day)	199.78	148.6	268.79	153.84
Flyby Altitude (km)	200	6347.3	300	11428
Earth TOF (day)	400	365.5	419.88	498.87
Flyby Altitude (km)	6000	5675	15026	300
Jupiter TOF (day)	1097.7	1056.1	1196	860
Arrival Velocity v_inf	4.43	4.56	3.87	5.77
TOF (day)	1697.5	1570.1	1884.7	1512.7
Prop. Mass Ratio	0.534	0.529	0.421	0.1786

exist in the bi-impulsive transfer. This is directly related to the physics of the problem since a low-thrust is supposed to produce a perturbation of the keplerian motion. The difference between the two becomes more and more evident as the number of spirals for the low-thrust arc are increased. In fact as the number increases the minima are shifted and the basins of attraction are stretched. It is interesting to notice that for a fixed value of the shaping parameter and for a fixed number of spirals the distribution remains unchanged. On the other hand the convergence analysis reveals that the shaping and the number of spirals play an important role in augmenting the number of local minima. Moreover the exponential sinusoid model introduces an additional difficulty since not all the transfers are physical. Non-physical transfers are regarded as unfeasible and for given values of the shaping parameter, number of spirals and transfer times no feasible solutions exist. This is understandable since for a given transfer time it is not physically possible to accommodate more than a certain number of spirals.

Table 15. EVEJ: Minimum Mass Problem

Parameter	Pruning+DE 1 400mN	Pruning+DE 2 300mN	DITAN 1 Min Mass	DITAN 2 Min Mass
Launch Date (MM-DD-YY)	June 27, 2010	June 27, 2018	June 21, 2010	July 01, 2018
Departure Velocity (km/s)	0.54119	1.118	0.54	1.1
Venus TOF (day)	199.78	148.6	291.09	154.3
Flyby Altitude (km)	200	6347.3	300	1147.0
Earth TOF (day)	400	365.5	409.55	501.5
Flyby Altitude (km)	6000	5675	9026	300
Jupiter TOF (day)	1097.7	1056.1	1024.13	812.42
Arrival Velocity v_inf	4.43	4.56	3.87	6.28
TOF (day)	1697.5	1570.1	1724.7	1468.2
Prop. Mass Ratio	0.534	0.529	0.327	0.138

Table 16. EVEJ: Comparison with the Literature

Parameter	STOUR-LTGA	IMAGO+EPIC	DITAN	Pruning+DE
Launch Date (MM-DD-YY)	May 09, 2015	Apr. 14, 2015	Apr. 21, 2015	Apr. 23, 2015
Departure Velocity (km/s)	2.0	1.8	1.8	0.713
Venus TOF (day)	119	184	177	185.76
Flyby Altitude (km)	4481	8816	2431	200
Earth TOF (day)	345	386	439	361.1
Flyby Altitude (km)	4219	5964	300	300
Jupiter TOF (day)	1027	901	888	1000
Arrival Velocity v_inf	5.97	4.53	5	6.23
TOF (day)	1491	1471	1504	1546
Prop. Mass Ratio	0.485	0.48	0.18	0.56

In the second part of the study we develop an incremental branch and prune algorithm that exploits problem information to reduce the search space. The resulting algorithmic complexity is quintic in the discretisation and quartic in the number of phases. Therefore though the algorithm is polynomial the exponent of the polynomial is significantly higher than for the MGA case. This is essentially due to the additional degree of freedom introduced by the shaping and thus by the low-thrust arc.

In the third part of the study the optimality of the solutions obtainable with the exponential sinusoid and with alternative methods has been analysed. The result of this analysis is that the solution of the exponential sinusoid is, in general, far from satisfying the necessary conditions for optimality. Nonetheless when it is used to initialise a direct approach it provides a correct estimation of the launch dates and of the encounter dates with the swingby planets. The non-optimality of the exponential sinusoid can be seen from the difference between the estimated propellant mass and the optimised one. On the other hand it should be underlined that when no Δv correction is required at the GA planet (i.e. the swingby is unpowered) the estimation of the propellant is close to the optimal value for the minimum control problem.

As an alternative to the exponential sinusoid we considered a shaping technique based on pseudo-equinoctial elements. This approach provides solutions that are closer to the optimal one than the exponential sinusoid, but the computational cost is considerably higher. However this approach can be significantly improved by speeding up the root-finding process to solve the time constraint. Furthermore the pseudo-equinoctial approach is more flexible at shaping low-thrust arcs and at accommodating constraints of different nature both on the thrust profile and on the boundary conditions.

We conclude with a conjecture. If the number of local optima is finite, the granularity of the sampling is sufficient to have a proper representation of the solution space and the pruning is

pushed to the limit, the algorithm will be able to reduce the search space to a neighborhood of the global solution or to a finite set of local solutions that contains the global one. Therefore, no matter the number of samples, since the algorithm is polynomial we can say that the solution of the problem can be obtained in polynomial time and therefore the complexity of the problem is polynomial.

References

1. D.H. Azimov and R.H. Bishop. Optimal transfer between circular and hyperbolic orbits using analytical maximum thrust arcs. In *AAS/AIAA Space Flight Mechanics Meeting*, AAS Paper 02-155, San Antonio, Texas, January 2002.
2. R.H. Bishop and D.H. Azimov. New analytical solutions to the fuel-optimal orbital transfer problem using low-thrust exhaust-modulated propulsion. In *AAS/AIAA Space Flight Mechanics Meeting*, AAS Paper 00-131, Clearwater, Florida, January 23-26 2000.
3. C. A. Coello Coello, D. A. Van Veldhuizen, and G. B. Lamont. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic Publishers, 2002.
4. V. Coverstone-Caroll, J.W. Hartmann, and W.M. Mason. Optimal multi-objective low-thrust spacecraft trajectories. *Computer Methods in Applied Mechanics and Engineering*, 186:387–402, 2000.
5. P. De Pascale, M. Vasile, and S. Casotto. Preliminary analysis of low-thrust gravity assist trajectories by an inverse method and a global optimisation technique. In *18th International Symposium on Space Flight Dynamics*, 11-15 October 2004, Munich, Germany.
6. K. Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*. Wiley, 2001.
7. T.J. Debban, T.T. McConaghy, and J.M. Longusk. Design and optimization of low-thrust gravity-assist trajectories to selected planets. In *AIAA/AAS Astrodynamics Specialists Conference*, AIAA Paper 2002-4729, Monterey, California, August 2002.
8. M. Dellnitz, O. Schütze, and T. Hestermeyer. Covering Pareto sets by multilevel subdivision techniques. *Journal of Optimization Theory and Applications*, 124:113–155, 2005.
9. Rimrott. F.P.J. and F.A. Salustri. Orbit dynamics and ward spirals. In *52nd International Astronautical Congress, Toulouse*, Paper IAF-01-A.6.09, 2001.
10. D. Izzo. Lambert's problem for exponential sinusoids. Technical report, ACT/ESTEC internal report, April 2005.
11. D. Izzo, V. M. Becerra, D. R. Myatt, S. J. Nasuto, and J. M. Bishop. Search space pruning and global optimisation of multiple gravity assist spacecraft trajectories. To appear in *Journal of Global Optimisation*, 2006.
12. S. Lee, P. von Allmen, A. E. Petropoulos W. Fink, and R. J. Terrile. Design and optimization of low-thrust orbit transfers using the q-law and evolutionary algorithms. In *IEEE Aerospace Conference*, March 2005.
13. Vasile M. Robust mission design through evidence theory and multi-agent collaborative search. 2005.
14. Vasile M. A hybrid multi-agent collaborative search applied to the solution of space mission design problems. In *International Workshop on Global Optimization*, 2005, Almeria, Spain.
15. N. Markopoulos. Non-keplerian manifestations of the keplerian trajectory equation, and a theory of orbital motion under continuous thrust. Paper AAS 95-217.
16. T.T. McConaghy, T.J. Debban, A.E. Petropoulos, and J.M. Longuski. An approach to design and optimization of low-thrust trajectories with gravity-assist. In *AAS/AIAA Astrodynamics Specialists Conference*, AAS Paper 01-468, Quebec City, Canada, July-August 2001.
17. D.R. Myatt, V.M. Becerra, S.J. Nasuto, and J.M. Bishop. Advanced global optimisation for mission analysis and design. ESA ARIADNA final report AO4532/18138/04/NL/MV, ESA, 2004.
18. De Pascale P. and Vasile M. Preliminary design of low-thrust multiple gravity assist trajectories. *Journal of Spacecraft and Rockets*, Vol. 43, No. 5:1065–1076, 2006.
19. P. Patel, D. Scheeres, and T. Zurbuchen. A shape based approach to spacecraft trajectories: Analysis and optimization. Paper AAS 05-130.
20. A. E. Petropoulos and J. M. Longuski. Shape-based algorithm for automated design of low-thrust. *Journal of Spacecrafts and Rockets*, 41, 2004.
21. A.E. Petropoulos. Some analytic integrals of the averaged variational equations for a thrusting spacecraft. Technical report, IPN Progress Report 42-150, August 15, 2002.
22. A.E. Petropoulos and J.M. Longuski. Shape-based algorithm for automated design of low-thrust, gravity-assist trajectories. *Journal of Spacecrafts and Rockets*, Vol. 41, No. 5, 2004.
23. A.E. Petropoulos, J.M. Longuski, and N.X. Vinh. Shape-based analytical representations of low-thrust trajectories for gravity-assist applications. In *AAS/AIAA Astrodynamics Specialists Conference*, AAS Paper 99-337, Girdwood, Alaska, August 1999.
24. G. Pinkham. Reference solution for low thrust trajectories. *Journal of the American Rocket Society*, Vol.32, No.5:775 76, May 1962.
25. K. Price, R. Storn, and J. Lampinen. *Differential Evolution - A Practical Approach to Global Optimization*. Springer, Berlin, 2005.
26. J.E. Prussing and V. Coverstone. Constant radial thrust acceleration redux. *Journal of Guidance Control and Dynamic*, Vol. 21, No. 3, 1998.
27. F.P.J. Rimrott and F.A. Salustri. On the ward spiral differential equation. *CSME Transactions*, 28(1):1–6, 2004.
28. O. Schütze, S. Mostaghim, M. Dellnitz, and J. Teich. Covering Pareto sets by multilevel evolutionary subdivision techniques. In C. M. Fonseca, P. J. Fleming, E. Zitzler, K. Deb, and L. Thiele, editors, *Evolutionary Multi-Criterion Optimization*, Lecture Notes in Computer Science, 2003.
29. A.R. Tanguay. *Space Maneuvers, Space Trajectories, Academic Press*. Academic Press, New York, 1960.
30. Ali M.M. Torn A. and Viitanen. Stochastic global optimization: Problem classes and solution techniques. *Journal of Global Optimization*, 14:437–447, 1999.

31. M. Vasile, F. Bernelli-Zazzera, N. Fornasari, and P. Masarati. Design of interplanetary and lunar missions combining low thrust and gravity assists. Final report of esa/esoc study contract no. 14126/00/d/cs, ESA/ESOC, 2001.
32. M. Vasile, L. Summerer, and P. DePascale. Design of earth-mars transfer trajectories using evolution-branching technique. *Acta Astronautica*, 56:705–720, 2005.
33. C. Yam, T. McConaghy, K. Chen, and J. Longuski. Preliminary design of nuclear electric propulsion missions to the outer planets. In *AIAA/AAS Astrodynamics Specialist Conference,Providence Rhode Island*, August 16-19 2004.