March 2025

Feasibility Assessment of Low-Thrust Multi-Target Trajectories via Sims-Flanagan Transcription

Mr. Yago Castilla Lamas[a*], **Dr. Joan-Pau Sánchez Cuartielles**[b], **Mr. José Carlos García Mateas**[c]

[a,b,c] *Institut Supérieur de l'Aéronautique et de l'Espace (ISAE-SUPAERO)*
[*] *Corresponding author: yago.castilla-lamas@student.isae-supaero.fr*

Abstract

Designing multi-target trajectories, such as asteroid tours, can potentially increase the scientific return of a mission. Additionally, its design in low-thrust is highly beneficial in terms of efficiency, but it imposes many difficulties regarding trajectory optimization. This problem was first addressed by the 4th GTOC edition, and its best-known solutions were designed as impulsive trajectories which then required careful and dedicated adjustments to find a feasible solution in low thrust. As a consequence, only a limited number of trajectories can be considered for conversion to low-thrust. This article presents a method to consistently perform this impulsive to low-thrust homotopy in an automated way. The algorithm generates a low-fidelity low-thrust solution, which is enough to assess the feasibility of the trajectory and is well suited to be used as an initial guess for a high-fidelity optimizer. The low-fidelity optimizer is based on the Sims-Flanagan transcription, and it has been implemented in MATLAB. The optimization algorithm uses the genetic algorithm (GA) implemented in MATLAB's Global Optimization Toolbox, along with a Monotonic Basin Hopping (MBH) algorithm to perform global search. This global search is then combined with local refinement via sequential quadratic programming (SQP). The input of the algorithm is only the list of ordered targets with a first guess for their flyby dates. Initially, an optimization of these flyby dates is performed using a one-impulse Sims-Flanagan transcription. Next, the arcs between targets are sequentially optimized in pairs. For each pair, several steps are performed. First, the flyby dates are re-optimized. Then, an optimal final velocity for the pair is computed to maximize the feasibility of the remaining trajectory. Finally, the classic multi-impulse Sims-Flanagan transcription is implemented and optimized to find a solution for the pair that respects the constraints while minimizing the difference with the computed optimal final velocity. The tool has been validated by converting the 45-asteroids winning solution of GTOC4, as well as other later solutions of up to 45 asteroids long. It has also been used to test over 200 tours, which showcase its suitability for evaluating a high number of trajectories. Furthermore, the one-impulse Sims-Flanagan transcription has proven to be a useful tool to efficiently estimate the feasibility of the trajectory with negligible computational cost, and it shows potential as a valuable tool for low-thrust multi-target trajectory design.

1. Introduction

Space exploration missions are traditionally designed to target a limited amount of visited celestial objects. However, recent missions show increasing complexity on its trajectories with several targets and celestial objects to be visited. These multi-target trajectories are of interest because they can greatly improve the scientific return of a mission. In addition, designing these missions in low-thrust is desirable to take advantage of the higher efficiency of these propulsion systems. However, the limited thrust of these systems together with the increased number of targets result in challenging problems regarding trajectory design and optimization.

The interest for this type of missions was clearly manifested in the subject of the 4th edition of the Global Trajectory Optimisation Competition (GTOC4) [1], which explored a low-thrust mission visiting the maximum number of near-Earth asteroids. The winning solution, submitted by the Moscow State University [2], as well as others well performing approaches [3–5], performed the selection of asteroids and the order in which they should be visited using impulsive trajectories. To approximate the low-thrust constraints, these approaches limited the magnitude of the provided impulses through different criteria. An example of such criteria is the acceleration parameter proposed by Barbee et al. [3]:

$$\alpha = \frac{\Delta v_{imp}/ToF}{T_{max}/m_0} \quad [1]$$

where Δv_{imp} is the impulse required to transfer from one target to the next one following a lambert arc, ToF is the time of flight between the two targets, T_{max} is the maximum thrust of the spacecraft, and m_0 is the mass of the spacecraft. According to this metric, an arc between targets would be feasible in low-thrust if the condition $\alpha \leq 1$ is satisfied.

These impulsive approaches can provide several hundred promising trajectories with dozens of targets. However, due to its impulsive nature, it cannot be guaranteed that they can be converted into feasible low-thrust trajectories. Moreover, even when conversion is possible, this low-thrust homotopy is not trivial and requires careful trajectory adjustments to find a feasible solution. Therefore, the number of these hundreds of trajectories that can be studied for conversion to low-thrust is very limited.

This article proposes a method to perform this low-thrust homotopy of multi-target trajectories in an automated way. The algorithm's output is a low-fidelity low-thrust trajectory that serves to quickly assess the feasibility of the trajectory and is well-suited to be used as an initial guess for a higher-fidelity optimization algorithm. Furthermore, the automation of the process significantly increases the number of trajectories studied for conversion to low-thrust, potentially increasing the number of available missions.

Olympio first proposed a method to address this problem in his work of 2011 [6]. He approached the optimization problem through an indirect method. These methods guarantee that the solution will satisfy the first order optimality conditions. However, they are highly sensitive to the initial guess of the costate variables, which lack physical significance, making it very difficult to estimate their order of magnitude or even their sign [7]. This article proposes a method to address the problem through a direct method. These methods are usually less accurate or provide sub-optimal solutions. However, they are generally more robust to the initial guess. The selected direct method is the Sims-Flanagan transcription, which is based on the discretization of the continuous thrust profile into a series of impulsive manoeuvers. The input of the algorithm is a list of ordered targets with a preliminary guess for their flyby dates. An optimization of both the flyby dates and the impulses is then performed to find a feasible trajectory that respects all the constraints.

The Sims-Flanagan transcription was first implemented with a local non-linear programming (NLP) algorithm. However, Yam et al. [8] showed that the use of global optimization algorithms to obtain an initial guess for the local trajectory optimizer, provided increased robustness to the solver. Furthermore, the work of Vasile et al. [9] on global optimization of space trajectories, showed better performance of the Monotonic Basin Hopping (MBH) algorithm over other global optimization algorithms to explore the design space. Yam et al. [10] then successfully applied this algorithm to the Sims-Flanagan transcription, showing its suitability for this problem. In the proposed method, global search is ensured through the implementation of a genetic algorithm (GA) and an MBH algorithm. Local refinement is performed using sequential quadratic programming (SQP).

This study has led to the development of a multi-target trajectory optimization MATLAB tool, which is available in a public GitHub repository that can be found in this work's references [11].

2. The Sims-Flanagan Transcription

The Sims-Flanagan transcription was first introduced by Sims & Flanagan in 1999 [12], as a direct method for low-thrust trajectory optimization. In this approach, the continuous thrust profile is discretized in several impulsive manoeuvres. Furthermore, the trajectory is divided in legs between two control points, that can be planets, asteroids or even free points in space. Each leg is then divided in N segments with an impulse in the middle of each segment. On each leg there is a match point. The trajectory is propagated forwards from the initial control node of the leg up to the match point, and backwards from the final control node of the leg to this match point. A constraint is imposed so that the state vector (position, velocity, and mass) at this match point is continuous. Therefore, the optimization variables of this transcription are the state vector at each control node and the intermediate impulses.

An additional constraint of the problem is that the mean thrust in a segment must be lower than the maximum thrust that can be provided by the spacecraft engine so that the trajectory is feasible. This mean thrust ($\overline{T}$) for a segment is defined as the mid-segment impulse magnitude divided by the time of flight (ToF) of the segment and multiplied by the mass of the SC (see equation 2). A diagram of the structure of this transcription is shown on Figure 1.

$$\overline{T} = \frac{\Delta V_{segment}}{ToF_{segment}} \cdot m_{SC} \qquad [2]$$

Since its introduction, the Sims-Flanagan transcription has been used to develop several software packages [13–15]. It was also used by Englander & Conway [16] to develop a fully automated solution for the low-thrust interplanetary mission design problem. In this method, the discrete dynamics, such as the number and order of flybys, were optimized by a heuristic optimizer outer loop. The inner loop trajectory optimization was then performed via the Sims-Flanagan transcription.

Regardless of all these existing different implementations of the transcription, to the knowledge of this author, it has never been successfully applied to trajectories with

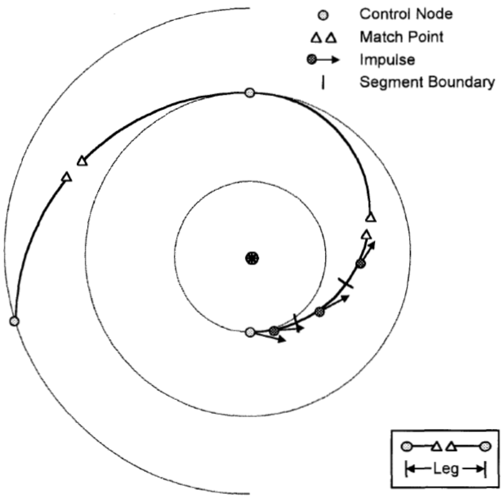


Fig. 1. Sims-Flanagan Transcription Structure [12]

more than a few targeted objects. Alemany & Braun [17] already suggested the use of the Sims-Flanagan transcription to design asteroid tours. However, and as they stated in this same work, the complexity of the problem with such a large number of flybys may pose difficulties to obtain a converged solution to the problem without a good initial guess. Section 2.1 will discuss the implementation of the transcription to this specific problem.

2.1 Sims-Flanagan Implementation

The implementation described in this article only addresses the case of simple flybys of the target objects. This means that the mass of these objects is sufficiently small for the effects of their gravity fields on the spacecraft's dynamics to be neglected. This assumption holds for asteroid tours such as the ones considered in GTOC4. Implementations that include rendezvous or gravity-assist manoeuvers are left for future studies. In this context, the trajectory consists of a sequence of legs connecting pairs of points in space. The positions of these points are determined by the ephemerides of the visited asteroids, according to their flyby dates.

To implement the Sims-Flanagan structure, a control node is placed at each of the flyby positions. Then, considering the original structure of the transcription, the optimization variables per leg would be the velocity vector at the initial and final nodes, and the N impulses of the leg. Moreover, three non-linear constraints per leg would be required, one for the match point in the middle of the leg, one for the maximum thrust of the propulsion system, and another one to ensure that the initial velocity matches the final velocity of the previous leg. Therefore, for a transcription of n_t targets and N impulses per leg, the number of design variables would be $(n_t{-}1) \times (N+2) \times 3$ with $(n_t{-}1) \times 3$ non-linear constraints.

To reduce the high dimension of this problem, the problem is split in several sub-problems. However, the most simple solution of dividing the problem in optimization problems of one leg at a time is not efficient. This is because the final state vector of one leg directly impacts the feasibility of the following one. Including several legs simultaneously in the optimization allows the algorithm to adapt the final state vector of the less demanded legs to ease the convergence of the most demanded ones. This is why the optimization is performed sequentially over optimization windows conformed by pairs of legs. At each step, this optimization window is shifted one leg. Figure 2 shows an illustration of this sequential process. At step i, the optimization window includes legs i-1 and i. Then, the results obtained for leg i-1 are frozen and the optimization window at iteration i+1 is shifted to include leg i plus the next leg i+1.

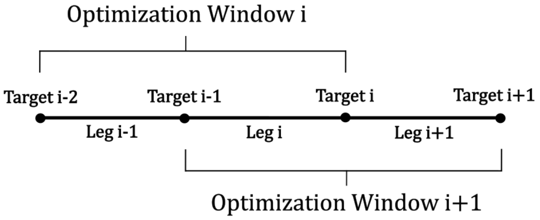


Fig. 2. Optimization Windows Structure.

Focusing on the implementation of the transcription to each optimization window, some changes are implemented with respect to the original transcription. The final position and velocity of the previous leg are used as the initial ones for the following one, and only forward propagation from the initial node is considered. The match point constraints are then applied at the final node of each leg. These constraints force the final positions resulting from the propagation of the legs to match the position of the targets. The advantage of this setup is that the constraints in the continuity of the state vector can be eliminated, as this continuity is guaranteed by the problem setup. Moreover, the number of design variables is reduced because the initial and final velocities of each leg are no longer explicit optimization variables, but the result of the propagation.

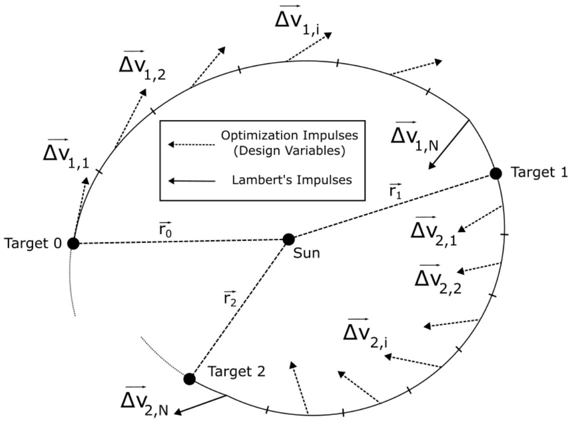
The explicit non-linear constraints on the match points can be removed if the final impulse of each arc is computed as a lambert arc. This way, the leg is propagated up to the point where this last impulse should be applied using the other N-1 impulses, which components are the explicit optimization or design variables. With the resulting position and velocity, the final last impulse is computed through the resolution of the lambert's problem of transferring the SC to the final node's position in the remaining ToF. This guarantees that the constraint in the position will be met and implicitly implements the non-linear constraint on the targets' flybys positions.

In addition, the mass of the spacecraft will be considered constant throughout each leg and equal to its initial value for that leg. The value is updated between legs based on the total impulse provided during the leg according to Tsiolkovski's equation (see equation 3). Finally, a uniform division of the ToF into N segments is considered to split each leg (see equation 4).

$$m_{i+1} = m_i \cdot e^{-\frac{\Delta V}{I_{sp} \cdot g_0}} \quad [3]$$

$$ToF_{segment} = \frac{ToF_{leg}}{N} \quad [4]$$

A diagram of the structure of this implementation is shown in Figure 3.



$\overrightarrow{\Delta v}_{1,i}$
$\overrightarrow{\Delta v}_{1,2}$
$\overrightarrow{\Delta v}_{1,1}$
Optimization Impulses (Design Variables)
Lambert's Impulses
$\overrightarrow{\Delta v}_{1,N}$
Target 1
$\overrightarrow{r}_1$
Target 0
$\overrightarrow{r}_0$
Sun
$\overrightarrow{\Delta v}_{2,1}$
$\overrightarrow{\Delta v}_{2,2}$
$\overrightarrow{r}_2$
$\overrightarrow{\Delta v}_{2,i}$
Target 2
$\overrightarrow{\Delta v}_{2,N}$

Fig. 3. Diagram of the Sims-Flanagan implementation structure applied to an optimization window.

The algorithm optimizes the final velocity of the optimization window v_{Fow} to minimize the difference with respect to an optimal final velocity v_{Fopt}. This optimal final velocity is computed so that the feasibility of the following legs is maximized. The method to compute its value is detailed in section 2.2.2. For each optimization window an optimization problem is to be solved in which the design variables are the cartesian components of each of the N-1 optimization impulses. The only remaining constraint is the one on the maximum mean thrust. The problem to be solved for each optimization window is then:

$$\min_{\substack{\{\Delta\mathbf{V_{1,1}};\ldots;\Delta\mathbf{V_{1,N-1}}; \\ \Delta\mathbf{V_{2,1}};\ldots;\Delta\mathbf{V_{2,N-1}}\}}} \quad J = ||\mathbf{v_{Fow}} - \mathbf{v_{Fopt}}||$$

$$\text{s.t.} \quad \max\left(\overline{T_{1,1}};\ldots;\overline{T_{1,N}};\overline{T_{2,1}};\ldots;\overline{T_{2,N}}\right) \leq T_{max} \quad [5]$$

$$\text{where} \quad \overline{T_{i,j}} = \frac{||\Delta\mathbf{V_{i,j}}||}{ToF_{segment}} \cdot m_{SC}$$

The direct application of the Sims-Flanagan transcription to the problem, resulted in a single optimization problem of $(n_t{-}1) \times (N + 2) \times 3$ design variables with $(n_t{-}1) \times 3$ non-linear constraints. Through this implementation, the problem is subdivided in $n_t - 1$ smaller problems of only $2 \times (N - 1) \times 3$ design variables and a single non-linear constraint.

2.2 One-Impulse Sims-Flanagan Transcription

The previously described implementation operates over fixed flyby dates of the targets. While the input of the method must provide an initial guess for these flyby dates, their values might be sub-optimal. As previously stated, most of the methods used to design tours rely on impulsive trajectory models. This is because these models require a negligible computational cost to evaluate the feasibility of a trajectory. In these models, the resolution of a lambert problem yields the departure velocity at each target required to reach the next target. The required impulses, which are applied at each flyby of the targets, are then simply computed as the difference between the arrival and departure velocities.

The algorithms used to design the tours typically use this transcription to optimize the flyby dates. However, the method described here introduces the use of a new model of impulsive transcription for this purpose. In this transcription, the impulses are no longer provided at the flybys, but instead they are provided in the middle of each leg. The state vector is then continuous at the flybys and the arrival velocity of a leg would be equal to the departure velocity of the next one. Given the arrival velocity to a target, the steps required to compute the mid-leg impulse of the next leg are:

1. Propagate the spacecraft trajectory for half of the ToF of the leg, with the initial velocity equal to the arrival velocity of the previous leg. Obtain the position vector, $\mathbf{r_{mid}}$, and the velocity vector, $\mathbf{v_{mid}}$, at this point.

2. Solve the lambert's problem to transfer the spacecraft from $\mathbf{r_{mid}}$ to the next target position, $\mathbf{r_{tg}}$, within the remaining half of the leg's ToF. This yields the velocity $\mathbf{v_{lamb}}$ required at $\mathbf{r_{mid}}$ to reach the target position in time.

3. The required impulse is then computed as the difference between the velocity vectors:

   $\Delta\mathbf{V_{mid}} = \mathbf{v_{lamb}} - \mathbf{v_{mid}}$

This transcription has shown to yield trajectories which better approximate a real low-thrust multi-target trajectory, when compared to the impulsive transcription with its impulses placed on the flybys. The observations sustaining this statement will be detailed in section 5.1. Moreover, it is possible to evaluate a trajectory via this transcription through the definition of the mean thrust in equation 2. This parameter is computed for each leg. For this computation, each leg is considered to be composed of a single segment, and the mass of the spacecraft is computed through the Tsiolkovski's equation, based on the mid-leg impulses magnitudes. According to this metric, a trajectory will be more feasible as the value of its maximum mean thrust is lower.

It should be highlighted that given an initial velocity and the flyby dates of the targets, the trajectory, required impulses, and mean thrusts, are fully determined. Therefore, the computational cost needed to evaluate a trajectory with this transcription remains negligible. This transcription is essentially a Sims-Flanagan transcription with one segment or impulse per leg. That is why it will be referred to as the one-impulse Sims-Flanagan (1-ISF) transcription.

2.2.1 Flybys Dates Optimization

Given the simplicity of this 1-ISF transcription, it can be easily used to perform an optimization of the targets' flyby dates. The flyby dates resulting from this optimization problem are taken as inputs by the multi-impulse Sims-Flanagan optimization problem described in section 2.1.

An initial optimization is first conducted over the full trajectory. The design variables of this optimization problem are the components of the initial velocity of the spacecraft $\mathbf{v_{launch}}$, and the vector of flyby dates of the targets $\mathbf{t_{flyby}} = [t_1, t_2, \ldots, t_{n_t}]$. The optimization will minimize the maximum mean thrust of the trajectory to maximize its feasibility. This results in an unconstrained optimization problem described as:

$$\begin{aligned} \min_{\{\mathbf{v_{launch}},\mathbf{t_{flyby}}\}} \quad & J = \max(\overline{T_1}, \overline{T_2}, \ldots, \overline{T_{n_t}}) \\ \text{where} \quad & \overline{T_i} = \frac{||\Delta\mathbf{V_{mid,i}}||}{ToF_i} \cdot m_{SC,i} \end{aligned} \qquad [6]$$

This optimization problem will be first solved for the full trajectory. Subsequently, it will be solved again after each optimization window iteration to re-optimize the flyby dates of the remaining trajectory. However, in these successive re-optimizations, the initial velocity won't be a design variable. Instead, its value will be fixed, matching the final velocity of the last computed leg. This re-optimization of the dates is required to account for the cumulated deviations of the actual low-thrust trajectory with respect to the initially computed 1-ISF transcription.

2.2.2 Optimal Final Velocity

The implementation of the Sims-Flanagan transcription described in section 2.1 optimizes the trajectory on each optimization window to minimize difference between the final velocity v_{Fow} and an optimal final velocity for that optimization window v_{Fopt}. This section addresses the way in which this optimal velocity is computed to maximize the feasibility of the following optimization windows. A valid approach would be to compute this velocity as the one required to have at the end of the optimization window i, to reach the target i+1 without providing any impulses on leg i+1. However, this will only maximize the feasibility of the following leg, without considering the feasibility of the rest of the trajectory.

The 1-ISF transcription is used to sustain the computation of this optimal final velocity with further information of its impact in the full remaining trajectory. Considering a fixed flyby dates vector $\mathbf{t_{flyby}}$, the optimal final velocity of the optimization window is computed so that the maximum mean thrust of the remaining legs, computed with the 1-ISF transcription, is minimized. Therefore, the value of the optimal final velocity for optimization window i, $v_{Fopt,i}$, is the result of the following unconstrained optimization problem:

$$\min_{\{\mathbf{v_{Fow,i}}\}} \quad J = \max(\overline{T_{i+1}}, \overline{T_{i+2}}, \ldots, \overline{T_{n_t}}) \qquad [7]$$

3. Trajectory Optimization

Section 2 has presented two different transcriptions with each own characteristic properties. The 1-ISF transcription provides a way of roughly assessing the low-thrust feasibility of a sequence of flybys with a negligible computational cost. However, it is not accurate, and it has no possibility of adapting the final conditions of the legs. The multi-impulse Sims-Flanagan transcription on the other hand is much closer to a real low-thrust trajectory, and it allows for the modification of the final conditions to improve the feasibility of subsequent legs. However, its computation requires solving a much more complex optimization problem and, with the described implementation, it does not allow for the optimization of the flyby dates. This section first presents the optimization algorithms used to solve optimization problems 5 to 7. Then, it describes the workflow of the algorithm and how the different transcriptions and optimization algorithms are put together in a unified structure to perform the low-thrust homotopy of the multi-target trajectory.

3.1 Optimization Algorithms

Three optimization algorithms are used in this work to solve the different described optimization problems. Depending on the dimension and structure of the optimization problem to be solved, the use of one algorithm is preferred over the others.

3.1.1 Sequential Quadratic Programming

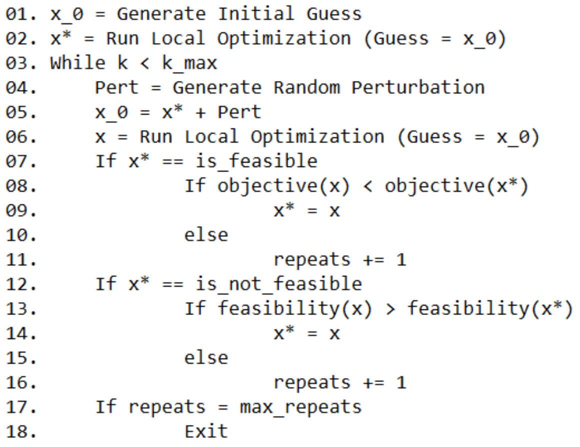
This method is a local search algorithm for constrained non-linear optimization. It is implemented via the MATLAB function *fmincon*. It is used both by itself but also to refine the solutions found by the global search algorithms. A maximum of 10^5 objective function evaluations is established to limit the optimization time.

3.1.2 Monotonic Basin Hopping

Pure local optimizers face a problem when dealing with low-thrust trajectories optimization. These problems have a great number of local minima, which makes that the local algorithms get ”stuck” in these local minima. Therefore, the results are very dependent on the initial conditions. MBH is a hybrid optimization algorithm that has proved to be efficient in solving this problem [9, 10].

The algorithm works by running successive local optimizations. After each local optimization, the solution found is randomly perturbed to get an initial condition for the following local optimization. This allows the algorithm to ”hop” the local minima and find a more optimal solution. This work uses a self-developed MBH algorithm built on MATLAB that uses *fmincon* as its local optimizer. The implementation also allows to handle constrained problems. For this, it first maximizes the feasibility of the solution. Once a feasible solution is found, it minimizes the objective function. Figure 4 shows the pseudocode of the algorithm. In this pseudocode, x is the result of the local optimization at each iteration and x* stores the best solution found up to that iteration.

There are two stop criteria for this algorithm with two associated parameters to be decided by the designer. The first one is that the algorithm reaches the maximum number of local optimization runs (k_{max}). The second one is that the algorithm reaches the maximum number of local optimizations without improvement of the best solution (max_{rep}).



```
x_0 = Generate Initial Guess
x* = Run Local Optimization (Guess = x_0)
While k < k_max
    Pert = Generate Random Perturbation
    x_0 = x* + Pert
    x = Run Local Optimization (Guess = x_0)
    If x* == is_feasible
        If objective(x) < objective(x*)
            x* = x
        else
            repeats += 1
    If x* == is_not_feasible
        If feasibility(x) > feasibility(x*)
            x* = x
        else
            repeats += 1
    If repeats = max_repeats
        Exit
```

Fig. 4. MBH Pseudocode.

Regarding the perturbing vector ϵ, it is generated following the scheme proposed by McCarty and McGuire [18]. Each component of this perturbing vector is generated as in equation 8. In this equation, r_i is a random number in the range [-1,1], x_i is the value of the corresponding component of the solution that is to be perturbed, and f is a parameter chosen by the designer that sets the maximum percentage of variation of each component. A value of 1 will represent a maximum change of 100% for each component. The value of each component of this perturbing vector will be directly added to the corresponding component of the current best solution.

$$\epsilon_i = \begin{cases} f r_i x_i, & \text{if } x_i \neq 0 \\ f r_i, & \text{if } x_i = 0 \end{cases} \qquad [8]$$

Three parameters must be chosen to run the MBH algorithm. The maximum number of local optimizations, k_{max}, the maximum number of successive local optimizations without improvement that are allowed before the stopping of the algorithm, max_{rep}, and the maximum percentage of change, f. These parameters must be tuned according to the specific problem at hand. In this work, k_{max} and max_{rep} have been fixed to 50 and 5 to have an acceptable time of optimization. Regarding f, it has been set to 0.15 as the result of several experiments to tune the algorithm. However, it has been observed that for the studied problems, the solution is not very sensitive to variations on this parameter.

3.1.3 Genetic Algorithm

A GA is a heuristic optimization technique based in the principles of the Darwin’s theory of the species evolution. In these methods, the different solutions of the problem are called individuals. Each of these solutions are defined by the value of its design variables, called the individual’s genes. Furthermore, the individuals are grouped in generations. In each generation, the solutions are evaluated according to the value of their objective function. The individuals with the highest objective function values are selected to mate with each other, creating a new set of individuals that form the next generation. Moreover, the algorithm randomly introduces mutations to the new generated offspring, which further expands the global search capabilities of this algorithm. On the other hand, the worst solutions are rejected so that the population is gradually improved.

In this work, the genetic algorithm is implemented using the *ga* function from MATLAB's Global Optimization Toolbox. This algorithm was chosen over other global optimization methods due to the authors' prior experience working with it. It offers greater global search capabilities than MBH but sacrifices precision in terms of optimality of the solution. Therefore, it is used when the global search capabilities of MBH are deemed insufficient, and its results are always refined afterward using the MBH algorithm.

3.2 Normalization

Proper normalization of optimization variables and the objective function typically improves the accuracy of results and the speed of optimization algorithms. The normalization described here is focused in optimization problem 5, as it is the one which shows the greatest complexity. However, a problem arises for this optimization problem, and it is that the optimization variables (cartesian components of impulses) do not have clear bounds, complicating the normalization. The solution adopted here involves a non-dimensionalization of the problem parameters. By analyzing the problem, reference magnitudes can be defined and used for this non-dimensionalization. These reference magnitudes are redefined for each optimization window.

In particular, three reference magnitudes are needed: longitude, time, and mass. As the design variables are impulses, which have dimensions of velocity, it would be interesting that the non-dimensionalization results in values of these impulses close to one. Therefore, such a reference velocity will be first chosen and then, the rest of reference magnitudes will be accordingly adjusted so that this reference velocity is maintained.

This reference velocity is chosen as the mean value between the maximum impulse of each leg in the initial guess for the optimization window. Then, the reference time is the defined as the mean ToF of the segments in the optimization window. Having defined reference values for velocity and time, the reference longitude is defined as the product of these two. Finally, the reference mass is defined as the initial mass of the spacecraft at the beginning of the optimization window.

$$\begin{cases} v_{ref} &= \frac{\max(\Delta V_{1,1};...;\Delta V_{1,N})\,+\,\max(\Delta V_{2,1};...;\Delta V_{2,N})}{2} \\ t_{ref} &= \frac{1}{N}\cdot\frac{ToF_{leg1}\,+\,ToF_{leg2}}{2} \\ r_{ref} &= v_{ref}\cdot t_{ref} \\ m_{ref} &= m_0 \end{cases} \qquad [9]$$

3.3 Method Structure

The described optimization problems must be solved using the appropriate optimization algorithms and in an ordered manner to sequentially optimize the different optimization windows. This part aims to provide a summarized description of the different steps performed at each optimization window. We assume that the algorithm just finished the optimization of the optimization window i containing legs i-1 and i (see Figure 2). It will next consider optimization window i+1 containing legs i and i+1. For this new optimization window, the results obtained for all previous legs up to leg i-1 are considered fixed.

First, optimization problem 6 is solved to re-optimize the flyby dates of the following legs. The optimized flyby dates correspond to target i+1 and all subsequent targets, but they do not affect the flyby date of target i or any previous targets. It should be remembered that for this general case of an intermediate leg in the trajectory, the initial velocity $\mathbf{v_{launch}}$ is not a design variable. Instead, it is equal to the final velocity of the previous optimization window i. This optimization problem is solved using the SQP algorithm because it does not present a complex structure.

Next, optimization problem 7 is solved to obtain the desired final velocity for the current optimization window $\mathbf{v_{Fopt,i+1}}$. The flyby dates which have just been computed are the ones considered for the computation of the one-impulse Sims-Flanagan transcription used to evaluate the objective function. This optimization problem has only the three cartesian components of the final velocity as design variables. Due to these reduced dimensions, the SQP algorithm is deemed sufficient to solve this optimization problem.

Lastly, optimization problem 5 is solved to obtain an optimal multi-impulse Sims-Flanagan transcription for the optimization window i+1 which minimizes the difference of its final velocity with respect to the computed optimal one, $\mathbf{v_{Fopt,i+1}}$. This problem is solved using the MBH algorithm. If this algorithm is not capable of finding a feasible solution, the GA is applied to perform a broader global search. In this case, the result obtained with the GA is subsequently re-optimized via the MBH algorithm.

4. 4th GTOC Edition Problem Description

The problem of low-thrust multi-target trajectories was addressed by the 4th edition of the Global Trajectory Optimization Competition (GTOC4) in 2009. More precisely, this edition dealt with the design of a low-thrust asteroid tour. The problem was formulated as follows: ”Maximize the relevance of a rendezvous mission to a given NEA by visiting the largest set of intermediate asteroids”. This problem has several constraints such as a given low-thrust propulsion system, a given mass of the spacecraft and propellant, a maximum departure excess velocity from earth, and a maximum mission duration of 10 years. The data for the problem can be found in Table 1. Two different trajectories proposed as solutions to this problem will be used to assess the developed algorithm.

ISP	3000 s
Max. Thrust	0.135 N
Max. Earth v_{inf}	4 km/s
SC Dry Mass	500 Kg
Propellant	1000 Kg

Table 1. GTOC4 Problem Statement Data [1].

4.1 Studied Trajectories

The two trajectories considered were initially designed as impulsive trajectories and later converted into low-thrust. This makes them an ideal test case for the developed algorithm, as it has already been proven that this conversion is feasible. The goal is to validate the developed algorithm by taking the impulsive trajectories and automatically performing the homotopy to an equivalent low-thrust trajectory.

The first trajectory that will be considered is the one proposed by the Moscow State University [2], which will be refer to as the Moscow’s solution. This solution, which won the competition, visits 45 asteroids. The team provides in the published article an impulsive solution visiting 50 asteroids. However, in the submitted solution 5 asteroids were removed from this solution to obtain the validated low-thrust trajectory. The authors do not provide any information on how the conversion to low-thrust was performed. The impulsive trajectory considered here to perform the low-thrust homotopy is the published one but removing the extra asteroids that were not included in the winning solution.

The second trajectory used to validate the algorithm is the one obtained by Gregory Phillip Johnson [4] in 2014, which will be referred to as Johnson’s solution. This trajectory is a later solution to the problem and visits 46 asteroids. Again, the author provides data for an impulsive trajectory, which is then converted into a low-thrust trajectory. This conversion is performed by first optimizing the flyby dates of the impulsive trajectory and then applying the indirect method developed by J. T. Olympio [6]. The data from the impulsive trajectory serves as input for the developed algorithm to attempt the low-thrust homotopy. Moreover, this trajectory has the advantage that the author also provides data for the low-thrust solution, allowing for a direct comparison with the results obtained using the method proposed in this work.

As the conversion of these trajectories to low-thrust has already been proven, the developed algorithm is expected to evaluate these trajectories as feasible or almost feasible to consider that it provides satisfactory results. The data about the visited asteroid names, flybys dates, and SC mass of the impulsive trajectories, can be found in Appendix A. This is the data which is taken by the algorithm as input.

In addition to these two trajectories, a database of more than 200 tours generated using the methodology proposed by García and Sánchez [5] is available for testing. However, these are all impulsive trajectories, and their conversion to low-thrust has not been proven. The algorithm is used to test them and attempt to identify new feasible low-thrust trajectories, leveraging the automation of the process to evaluate a large number of tours.

5. Results and Discussion

The algorithm is first applied to the Moscow’s solution. Figure 5 shows the obtained mean thrust profile along the trajectory. The red dashed lines in this figure mark the asteroids flybys, and the blue dashed line represent the maximum available thrust of the spacecraft. Therefore, for the trajectory to be feasible, the mean thrust should remain below this line.

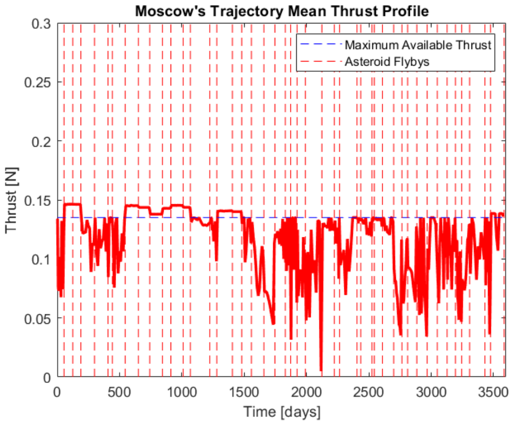


Fig. 5. Moscow’s solution mean thrust profile.

The obtained mean thrust profile shows an almost feasible trajectory in terms of thrust. Some legs are above the maximum available thrust. However, this value is only exceeded by a maximum of 0.11 N, which represents an 8% of the maximum thrust. It is expected that a dedicated low-thrust trajectory optimization software would be able to reduce this thrust below the maximum allowed. It should be noted that this implementation does not satisfy the rendezvous at the last asteroid stated by the GTOC4 subject because, in this implementation, only flybys are considered. The velocity difference in the last flyby has a not negligible value of 2.75 km/s. However, the implementation developed here considers a fixed value of the last flyby date equal to the one provided in the initial guess. This last phase would need to be studied in detail to assess if the feasibility of the rendezvous with the last asteroid could be improved by shifting the date of the rendezvous. When comparing the optimized flyby dates with those of the impulsive trajectory, the maximum shift in a flyby date from its original value is 5.02 days, while the mean shift is 1.2 days.

This solution is almost feasible in terms of maximum thrust. However, its feasibility does not hold in terms of propellant mass. Figure 6 shows the mass evolution of the spacecraft along the trajectory. This value is compared to the mass provided by the impulsive trajectory used as initial guess.

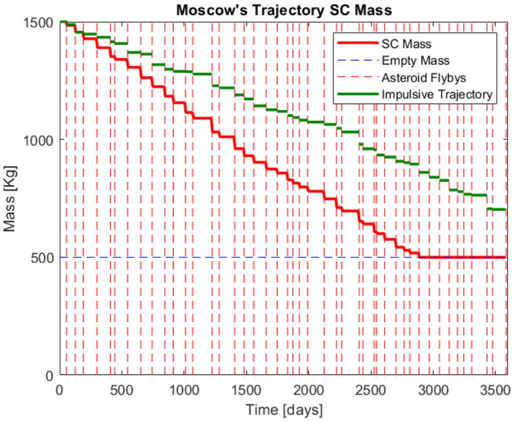


Fig. 6. Moscow's solution spacecraft mass evolution.

It can be observed how the SC mass for the obtained trajectory is expended much faster than for the impulsive trajectory. This results in the spacecraft consuming all its propellant by the 36th leg. For the last 9 legs, the spacecraft mass is maintained at the 500 kg empty mass so that the thrust value is still representative.

This is not considered very concerning because the optimization of the final mass is not considered at any point in the proposed method. Moreover, observing the thrust profile in Figure 5, it is clear that this is a suboptimal trajectory because it does not present the optimal bang-bang structure predicted by the primer vector theory. Therefore, re-optimizing the trajectory to maximize the final mass should provide a clear improvement in this sense. Figures 7 and 8 show the results of an attempt to perform this re-optimization.

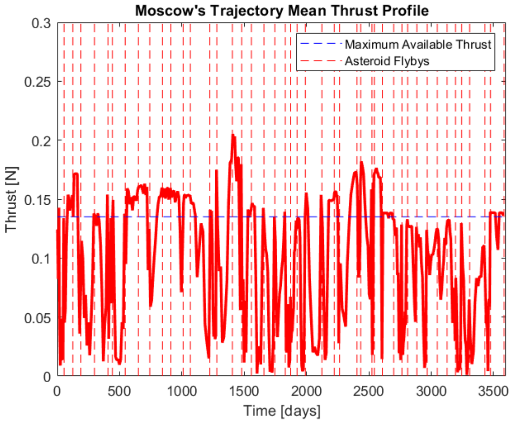


Fig. 7. Moscow mass re-optimized solution. Mean thrust profile.

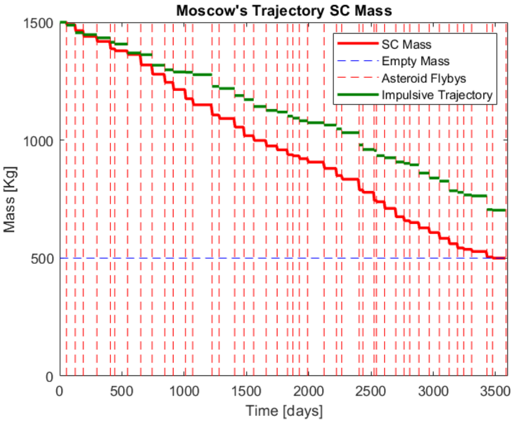


Fig. 8. Moscow mass re-optimized solution. SC mass evolution.

In this case, the re-optimized thrust profile shows a structure much closer to the expected bang-bang behaviour. Unfortunately, a good management of the maximum thrust constraint has not been achieved in this re-optimization, which leads to a more significant infeasibility in terms of thrust. However, the spacecraft mass holds now above the empty mass for the full trajectory, except for the final leg. Even if this solution is not feasible, it shows the potential of re-optimizing solutions of the algorithm to meet the mass constraint. Again, it is expected that a dedicated low-thrust trajectory optimization software would be able to convert the results of this algorithm in a feasible trajectory both in terms of thrust and mass.

Next, the algorithm is applied to try and convert the Johnson's solution. Figure 9 shows the obtained mean thrust profile.

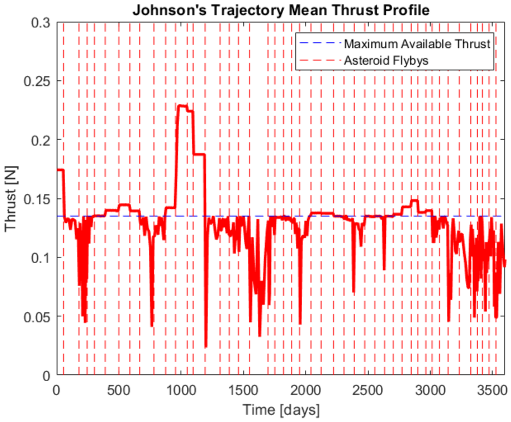


Fig. 9. Johnson's solution mean thrust profile.

In this case, it cannot be considered that the algorithm has achieved a successful conversion. The obtained solution presents several legs in which the thrust greatly exceeds the maximum available one, with a maximum exceedance of 70% over the maximum thrust. On the other hand, this solution can be considered to satisfy the final rendezvous condition, as the difference between the spacecraft final velocity with respect to the one of the last asteroid is just 1 m/s. Similarly to the solution obtained for the Moscow's trajectory, this solution is not feasible in terms of mass. The spacecraft runs out of propellant after the 31st leg. However, the thrust profile does not present a bang-bang structure and a re-optimization of the trajectory maximizing the final mass should provide significant improvements in this sense. In this case, the maximum shift on a flyby date compared to the impulsive solution is 17.7 days, but the mean value of these shifts is 1.57 days, similar to the value obtained for the Moscow's trajectory. It is interesting to note that these values of the flyby dates shifts are greater than those observed between the impulsive solution and the low-thrust solution obtained through Olympio's method. In that case, the maximum shift were 4.52 days and the mean value only 0.67 days.

The infeasible legs of the results for this trajectory include the first leg and legs 12, 13 and 14. The first leg should not pose significant problems. This is because the Earth departure date has been considered fixed in the implementation. Moreover, the Earth v_∞ is only 1.07 km/s, well below the maximum possible one, which is 4 km/s. Adjusting these two parameters, it should be possible to reduce the thrust of this first leg to a feasible value. For intermediate legs 12, 13 and 14, it is interesting to study how removing one of the asteroids involved in these legs would affect the feasibility of the solution. These are asteroids 11 to 14. Four new trajectories with 45 asteroids are created by removing on each of them a different asteroid among these four. Each solution is then run through the algorithm. It is found that removing asteroids 13 or 14 do not result in an improved feasibility of the solution. However, when asteroids 11 or 12 are removed, the feasibility problem of these 3 legs is solved and the algorithm converges to a solution in which the thrust for these three legs stays just above the maximum thrust limit. Therefore, it is clear that the most critical leg of the trajectory is leg number 11. The infeasibility of this leg then propagates to subsequent legs because, as a result of this infeasibility, the SC cannot adjust the final conditions of the leg. Figure 10 shows the thrust profile obtained for the solution in which asteroid number 12 have been removed.

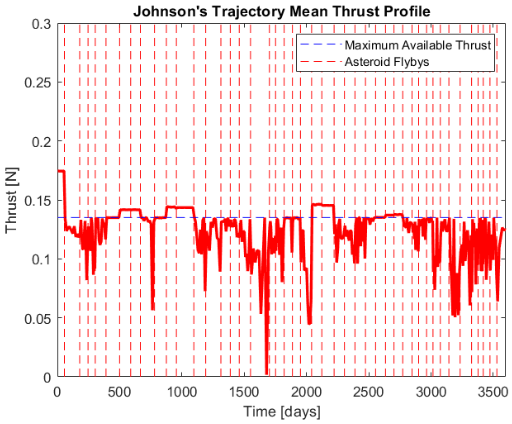


Fig. 10. Johnson's solution mean thrust profile. Asteroid number 12 removed.

Finally, the algorithm has been applied to automatically try to perform the low-thrust homotopy of more than 200 tours generated with García and Sánchez's dynamic programming method, including tours of up to 50 asteroids. However, the algorithm was not able to find feasible thrust profiles for them. Figure 11 shows the mean thrust profile obtained for one of the more promising trajectories.

This trajectory shows a large infeasible leg interval comprising legs 6 to 13. A similar test to the one performed for Johnson's solution was conducted, removing asteroids from the trajectory. It was found that the best results were obtained when asteroid number 7 was removed.

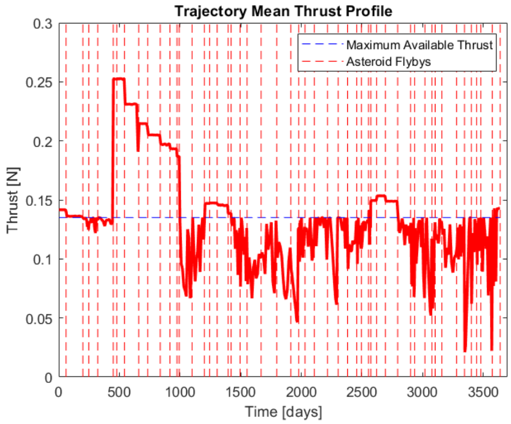


Fig. 11. Mean thrust profile of a promising tour generated by García and Sánchez's dynamic programming algorithm.

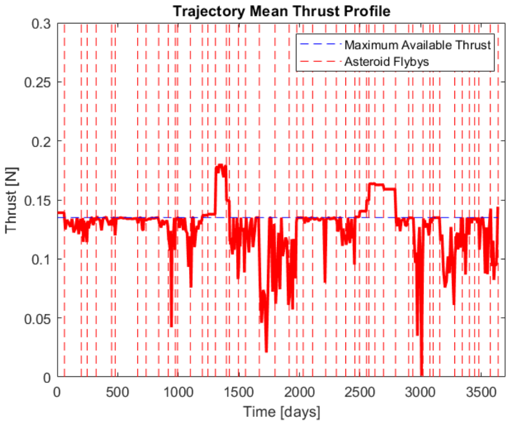


Fig. 12. Mean thrust profile of a promising tour generated by García and Sánchez's dynamic programming algorithm. Asteroid number 7 removed.

In this case, even though only one asteroid was removed, the entire 8-leg interval became feasible. Figure 12 shows the mean thrust profile obtained for this shortened trajectory. Note that for both this trajectory and Johnson's, removing a single asteroid from the initial legs of the infeasible interval made the entire interval feasible. This seems to confirm that the infeasibility of a single leg can propagate to many adjacent legs. In this case, the propagation mainly occurred to subsequent legs rather than backward to previous legs. However, more research is needed to determine whether this is always the case.

5.1 1-ISF Transcription Assessment

This section will assess the convenience of using the one-impulse Sims-Flanagan transcription compared to the classical impulsive transcription where the impulses are placed at the flybys. This last transcription will be referred to simply as the impulsive transcription. This choice is sustained on one main observation made with a previous and more primitive implementation of this algorithm. In this version, the algorithm optimized each leg individually, minimizing the maximum thrust of that leg. This approach was deemed insufficient to converge to feasible trajectories mainly because each optimization problem does not have any information on how its final velocity will impact the feasibility of the following arcs. Therefore, the less demanded legs are not able to help in the convergence for the more demanded ones, which results in diverging thrust profiles with increasing magnitude along the trajectory. However, a significant correlation was observed between this thrust profile and the one of the 1-ISF transcription. This is shown is Figures 13 and 14, which correspond to Moscow and Johnson's trajectories, respectively. In these figures, the blue lines correspond to the mean thrust on each leg computed through the impulsive transcription. The red lines are the mean thrust for each leg but computed with the 1-ISF transcription. Finally, the black lines show the maximum mean thrust in each leg obtained with the approach that optimizes each leg individually, minimizing the maximum thrust of each leg.

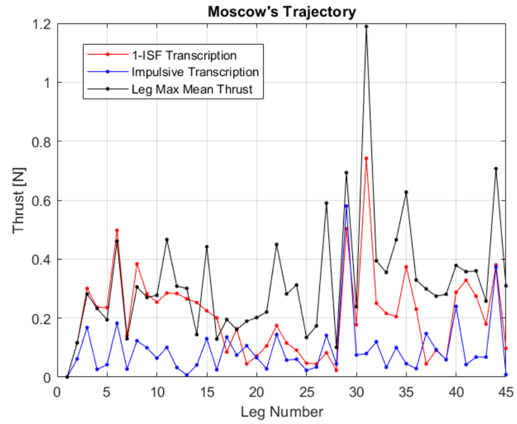


Fig. 13. Moscow's trajectory. 1-ISF transcription, impulsive transcription and maximum mean thrust.

It can be observed how for both trajectories, the curve of the 1-ISF transcription follows much closely the line of the maximum mean thrust, when compared to the impulsive transcription. Among the trajectories over which the algorithm has been tested, the Moscow solution is the one in which greater difference have been observed between the computed maximum mean thrust profile and the one predicted by the 1-ISF transcription. In this trajectory,

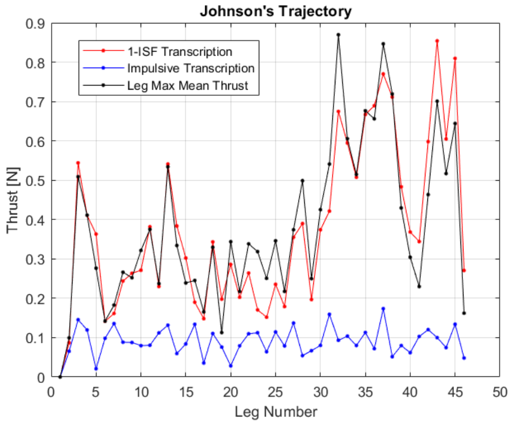


Fig. 14. Johnson's trajectory. 1-ISF transcription, impulsive transcription and maximum mean thrust.

there exists two leg intervals - legs 17 to 29 and 37 to 40 - in which the 1-ISF transcription estimation present a high discrepancy with the obtained maximum mean thrust for those legs. However, in these intervals, the estimation can be considered at least as good as the one provided by the impulsive transcription. For the rest of the trajectory, the 1-ISF transcription seems to provide a better estimation of the required thrust. On Johnson's trajectory, the estimated thrust using the 1-ISF transcription is clearly more accurate than the impulsive transcription for the full trajectory. Based on these results, the thrust estimated by this 1-ISF transcription seems to be a better approximation for the low-thrust multi-target trajectory than the impulsive transcription.

It should be remembered that this low-thrust profile which seems to be approximated by the 1-ISF transcription, was obtained minimizing the maximum thrust of each arc individually. Then, if the full 1-ISF transcription of a trajectory predicts a feasible thrust profile, it should be possible to convert this trajectory to low-thrust by only minimizing the maximum thrust of each leg without any further considerations. As the estimation is far from being exact, this might not be the case. However, such a trajectory should, in any case, be easier to convert into low-thrust following the method proposed in this work.

It is interesting to observe how the two transcriptions evolve as the flyby dates are optimized by the algorithm. Figure 15 shows the mean thrust computed with both transcriptions for the Johnson's solution. The red and blue lines with dot markers show the thrust obtained with the respective transcription for the final optimized flyby dates obtained with the proposed algorithm. The green and light blue lines with cross markers represent the thrust obtained with these transcriptions but for the flyby dates that were inputted as initial guess. Finally, the black line shows the actual maximum thrust obtain on each leg for the thrust profile obtained for this trajectory, which was shown in Figure 9.

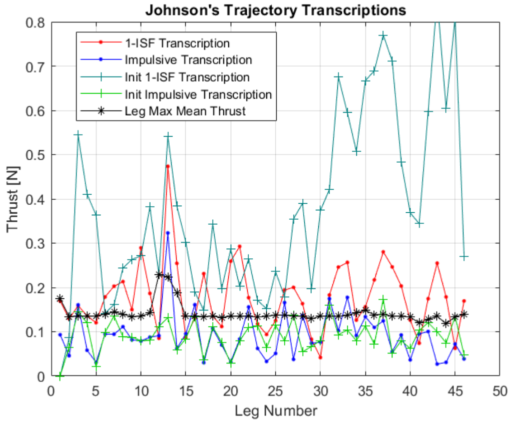


Fig. 15. 1-ISF and impulsive transcription for the initial guess and final flyby dates and leg's maximum thrust. Johnson's trajectory.

It is observed how for the initial flyby dates, the impulsive transcription results in mean thrust values mostly under the maximum thrust. On the other hand, the 1-ISF transcription shows extremely high values for the thrust of up to 0.85 N. Once the optimization of the flyby dates is performed, the thrust obtained with the 1-ISF transcription is greatly reduced for most of the trajectory. The maximum thrust is now 0.47 N and it is obtained in leg number 13, which is precisely in the infeasible leg interval that presented a thrust higher than the available one. With these new dates, the impulsive transcription also presents the spike on leg number 13. It should be noted that both the 1-ISF transcription and the impulsive transcription estimate that several legs would be infeasible, as the estimated mean thrust is above the maximum available one. However, these legs were then converted to feasible low-thrust legs. This shows the ability of the algorithm to adapt the final conditions of previous legs to ease the convergence of the following ones. It should also be noted that the 1-ISF transcription shows in almost every leg higher values for the thrust than the impulsive transcription, which are in most of the arcs above the thrust limit. It would seem then that the 1-ISF transcription generally overestimates the required thrust. This is another fact that leads to think that if this transcription estimates that a trajectory is feasible, the low-thrust homotopy to a feasible low-thrust trajectory should be possible.

5.2 Limitations and Further Works

It should be mentioned that the implementation of the algorithm has shown robustness problems in the sense that two executions over the same initial guess can lead to significantly different results. These differences can result in the algorithm predicting that a trajectory is feasible on the first run but if a second execution is run over the same initial guess, the algorithm's output might be an infeasible trajectory. As an example, Figure 16 shows a result that was obtained for the Moscow solution, with the same initial guess as the one used to obtain the results shown in Figure 5.

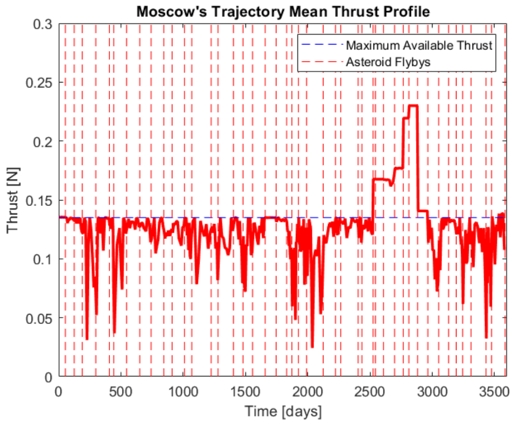


Fig. 16. Infeasible Moscow's solution.

While this is an extreme case, it perfectly highlights the robustness problems of the implementation. These most certainly come from the randomness introduced by the MBH and genetic algorithm used to solve optimization problem 5, associated to the optimization of the multi-impulse Sims-Flanagan transcription. These algorithms appear to not sufficiently explore the design space and so they converge to different local minima between different executions of the algorithms. This results in different final conditions on the legs which in the end accumulates and completely change the resulting trajectory. More work would be required to better implement and solve this optimization problem to achieve more consistent results. A promising approach may be to change the design variables from the cartesian components of the impulses to a description with a magnitude and two angles to orient them with respect to a given reference frame. This might prove beneficial because this parametrization presents more clear bounds for each optimization variable.

Further research may also address the study of the automatic removal of asteroids from infeasible legs in the most optimal way. This might require a deeper analysis of how the infeasibility of one leg propagates to adjacent legs in order to predict which asteroid has the greatest impact on feasibility and should therefore be removed from the solution.

Finally, understanding which characteristics of the initial and final asteroids make a leg critical, could help predict which impulsive trajectory legs may be challenging to convert to low-thrust. One potential factor might be the relative velocity between the spacecraft and the asteroid, as higher relative velocities limit flexibility in adjusting the flyby date, negatively impacting feasibility. However, analysis shows that in the Johnson's trajectory, asteroids 11 and 12, which are problematic, have lower relative velocities than asteroids 13 and 14, suggesting this is not a decisive factor. Another possible influence is the asteroid's orbit inclination, as higher inclinations could require more demanding out-of-plane maneuvers when adjusting the flyby date. However, asteroid 7, which provided the best results when removed from García and Sánchez's solution, has the lowest inclination among the asteroids involved in the infeasible leg interval. Therefore, this variable also does not seem to have a significant isolated impact when determining whether a leg will be feasible. Further studies could explore whether the combination of these two factors, or other leg or asteroid parameters, could help predict the feasibility of the legs.

6. Conclusions

An algorithm to automatically perform low-thrust homotopy of impulsive multi-target trajectories has been implemented. This algorithm sequentially optimizes pairs of trajectory legs. First, it uses a one-impulse Sims-Flanagan transcription to optimize the flyby dates of the targets and compute an optimal final velocity for each optimization window. Then, a multi-impulse Sims-Flanagan transcription is implemented and optimized to achieve the desired final conditions.

The algorithm has been validated on two asteroid tour trajectories proposed for GTOC4: the 45-asteroids Moscow trajectory and the 46-asteroids Johnson trajectory. It has also been tested on more than 200 asteroid tours generated via dynamic programming, showcasing its suitability for evaluating a high number of trajectories.

The algorithm successfully converted the Moscow solution into an almost feasible low-thrust trajectory in terms of maximum thrust. However, the resulting trajectory was not feasible in terms of propellant mass, as the spacecraft would run out of propellant by the 36th leg. A re-optimization of the trajectory ensured sufficient propellant remained available up to the final leg. In this new solution, the thrust profile exhibited a structure closer to the expected optimal bang-bang behavior, but the thrust limit was exceeded due to improper handling of the optimization constraints.

For the Johnson trajectory, the results showed three intermediate legs where the thrust significantly exceeded the maximum available thrust. Removing the second asteroid involved in this leg interval allowed the algorithm to converge to an almost feasible solution, visiting 45 asteroids. A similar behavior was observed in one of the most promising trajectories generated by García and Sánchez's method. In this case, removing a single asteroid from the early infeasible legs of an 8-leg long infeasible interval, led to the resolution of feasibility issues for the entire interval. This shows how the infeasibility of a single leg can be propagated to many adjacent trajectories and highlights the importance of further studies to determine which asteroid has the greatest impact on feasibility, so that it can be removed from the trajectory.

Finally, the one-impulse Sims-Flanagan transcription has proven to be a good approximation for a low-thrust multi-target trajectory where each leg is individually optimized to minimize its maximum thrust. In general, this transcription tends to yield higher thrust values than the more classical impulsive model. These observations suggest that if the one-impulse Sims-Flanagan transcription of a trajectory is feasible, the low-thrust homotopy to a feasible trajectory should also be possible. Although further work is required to confirm this, the one-impulse Sims-Flanagan transcription appears to be a promising tool for low-thrust multi-target trajectory design.

Appendix A (GTOC4 Trajectories)

No.	t [MJD]	m [kg]	Asteroid
0	58676.40	1500.00	Earth
1	58731.65	1487.47	2006QV89
2	58801.45	1455.53	2006XP4
3	58866.95	1447.41	2008EP6
4	58973.95	1433.87	2007KV2
5	59084.55	1415.27	2005XN27
6	59119.45	1407.21	2006TB7
7	59221.75	1369.68	2008AF4
8	59326.95	1362.41	2006HF6
9	59419.55	1317.69	2008PK3
10	59520.05	1298.40	2007VL3
11	59586.25	1289.05	2006AN
12	59685.15	1287.81	2006UQ216
13	59746.75	1277.76	2006KV89
14	59901.25	1227.92	154276
15	59955.85	1219.21	2006AU3
16	60078.75	1189.08	2008GM2
17	60157.05	1172.01	2008NA
18	60233.95	1142.62	2005CD69
19	60334.75	1126.08	2008KE6
20	60422.75	1119.34	2007VD184
21	60507.25	1101.87	2008EQ
22	60549.75	1093.41	2003YP3
23	60601.15	1082.33	2005WK4
24	60665.25	1073.82	2004QJ13
25	60796.75	1064.09	2006UB17
26	60897.85	1047.89	1993FA1
27	60938.15	1031.98	2005EU2
28	61082.15	979.77	143527
29	61110.35	960.43	2005GY8
30	61201.65	955.95	199801
31	61221.35	934.20	2005XW77
32	61285.55	925.47	2008AP33
33	61377.65	907.40	2004JN1
34	61441.75	901.59	2007US51
35	61486.75	895.42	2008QB
36	61562.75	860.65	1995SA4
37	61646.75	839.49	2006WR127
38	61727.45	826.31	2002CW11
39	61806.65	785.42	2003WP25
40	61867.75	778.96	2006BZ147
41	61921.85	767.85	2004RN111
42	61979.55	763.83	2006VP13
43	62105.85	705.79	2008RH1
44	62151.65	703.19	2007HW4
45	62261.25	616.45	2000SZ162

Table 2. Moscow's solution [2].

No.	t [MJD]	m [kg]	Asteroid
0	59466.01	1500.00	Earth
1	59521.45	1500.00	2007VL3
2	59646.51	1475.03	2005CD69
3	59712.87	1445.24	164207
4	59768.40	1425.32	2003LN6
5	59856.87	1419.93	2001SQ3
6	59963.48	1387.41	2003YG136
7	60051.66	1349.72	2002GR
8	60132.93	1328.15	186844
9	60248.09	1296.64	2006UB17
10	60340.42	1274.68	2002AU4
11	60421.69	1255.34	2005EB30
12	60513.76	1224.11	163364
13	60561.85	1205.37	2004FD
14	60657.28	1188.57	2008GF1
15	60778.42	1157.35	2008PW4
16	60852.23	1127.47	138175
17	60925.93	1119.86	2002EM7
18	61014.46	1089.48	2000SZ162
19	61161.70	1054.68	2005EZ169
20	61217.45	1050.13	2008JP24
21	61287.36	1034.04	2000AA6
22	61350.13	1013.53	1997UA11
23	61416.64	991.23	2006KV89
24	61508.61	974.09	2000EB14
25	61586.09	947.95	2004JO20
26	61685.62	924.96	2003OT13
27	61774.17	887.88	2006VU2
28	61837.15	878.08	2006XP4
29	61940.03	858.12	2006GC1
30	62020.65	839.36	2003XK
31	62104.66	798.21	2006RJ1
32	62171.01	780.54	1998UY24
33	62241.09	759.72	175706
34	62311.30	743.73	2008CP
35	62367.10	725.61	2006HF6
36	62429.73	712.75	2008LG2
37	62481.67	686.30	2004PR92
38	62537.42	678.12	2001RQ17
39	62611.49	661.37	2001XG1
40	62700.21	645.89	2004BN41
41	62792.83	617.95	2004UT1
42	62844.78	600.03	2008AP33
43	62886.17	588.39	2006QK33
44	62941.61	576.69	2002TX59
45	62993.56	556.94	2006VY2
46	63117.74	525.32	2006QQ56

Table 3. Johnson’s solution [4].

References

[1] R. E. Régis Bertrand and B. Meyssignac, "Problem description for the 4th global trajectory optimisation competition," Centre National d'Etudes Spatiales (CNES), 2009.

[2] I. S. Grigoriev and M. P. Zapletin, "Choosing promising sequences of asteroids," *Automation and Remote Control*, vol. 74, No. 8, pp. 1284–1296, 2013.

[3] B. W. Barbee, G. W. Davist, and S. H. Diaz, "Spacecraft trajectory design for tours of multiple small bodies," *Advances in the Astronautical Sciences*, vol. 135, No. 3, pp. 2169–2188, 2009.

[4] G. P. Johnson, *A Tabu Search Methodology for Spacecraft Tour Trajectory Optimization*, 1st ed. Texas: University of Texas, 2014.

[5] J. C. G. Mateas and J. P. S. Cuartielles, "Low-thrust automated asteroid tour trajectory design via dynamic programming," in *AAS 23-217, AAS/AIAA Astrodynamics Specialist Conference*, Big Sky, Montana, Aug. 13–17, 2023.

[6] J. T. Olympio, "Optimal control problem for low-thrust multiple asteroid tour missions," *Journal of Guidance, Control, and Dynamics*, vol. 34, No. 6, pp. 1709–1720, 2011.

[7] B. A. Conway, *Spacecraft Trajectory Optimization*, 1st ed. 32 Avenue of the Americas, New York: Cambridge University Press, 2010.

[8] C. H. Yam, F. Biscani, and D. Izzo, "Global optimization of low-thrust trajectories via impulsive delta-v transcription," in *27th International Symposium on Space Technology and Science*, Tsukuba, Japan, Jul. 5–12, 2009.

[9] M. Vasile, E. Minisci, and M. Locatelli, "Analysis of some global optimization algorithms for space trajectory design," *Journal of Spacecraft and Rockets*, vol. 47, No. 2, pp. 334–344, 2010.

[10] C. H. Yam, D. D. Lorenzo, and D. Izzo, "Constrained global optimization of low-thrust interplanetary trajectories," in *IEEE Congress on Evolutionary Computation*, Barcelona, Spain, Jul. 18–23, 2010.

[11] Y. C. Lamas. "Github repository: Gtoc4 sims-flanagan low-thrust homotopy." (accessed 15.03.25). (Mar. 30, 2025), [Online]. Available: `https://github.com/Yagocastilla/RP_Sims_Flanagan`.

[12] J. A. Sims and S. N. Flanagan, "Preliminary design of low-thrust interplanetary missions," in *AIAA/AAS Astrodynamics Specialist Conference*, Girdwood, Alaska, Aug. 15–19, 1999.

[13] T. T. McConaghy, T. J. Debban, A. E. Petropoulos, and J. M. Longuski, "Design and optimization of low-thrust trajectories with gravity assists," *Journal of spacecraft and rockets*, vol. 40, No. 3, pp. 380–387, 2003.

[14] J. A. Sims, P. A. Finlayson, E. A. Rinderle, and M. A. Vavrina, "Implementation of a low-thrust trajectory optimization algorithm for preliminary design," in *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, Keystone, Colorado, Jun. 21–24, 2006.

[15] M. T. Ozimek, J. F. Riley, and J. Arrieta, "The low-thrust interplanetary explorer: A medium-fidelity algorithm for multi-gravity assist low-thrust trajectory optimization," in *AAS/AIAA Space Flight Mechanics Meeting*, Maui, Hawai, Jan. 13–17, 2019.

[16] J. A. Englander and B. A. Conway, "Automated solution of the low-thrust interplanetary trajectory problem," *Journal of Guidance, Control, and Dynamics*, vol. 40, No. 1, pp. 15–27, 2017.

[17] K. Alemany and R. D. Braun, "Survey of global optimization methods for lowthrust, multiple asteroid tour missions," in *AAS/AIAA Space Flight Mechanics Meeting*, Sedona, Arizona, Jan. 28–Feb. 1, 2007.

[18] S. L. McCarty, L. M. Burke, and M. McGuire, "Parallel monotonic basin hopping for low thrust trajectory optimization," in *AAS/AIAA Space Flight Mechanics Meeting*, Kissimmee, Florida, Jan. 8–12, 2018.