POLITECNICO DI TORINO
Repository ISTITUZIONALE

Hybrid Optimization Technique for Finding Efficient Earth–Moon Transfer Trajectories

Original
Hybrid Optimization Technique for Finding Efficient Earth–Moon Transfer Trajectories / Casalino, L., D'Ottavio, A., Fasano, G., Pintér, J.D., Roberto, R.. - In: ALGORITHMS. - ISSN 1999-4893. - 19:1(2026). [10.3390/a19010080]

Availability:
This version is available at: 11583/3006869 since: 2026-01-23T08:37:29Z

Publisher:
MDPI

Published
DOI:10.3390/a19010080

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

11 September 2026

algorithms

MDPI

Article

Hybrid Optimization Technique for Finding Efficient Earth–Moon Transfer Trajectories

Lorenzo Casalino [1,*], Andrea D'Ottavio [2], Giorgio Fasano [2], Janos D. Pintér [3] and Riccardo Roberto [1]

[1] Department of Mechanical and Aerospace Engineering, Politecnico di Torino, 10129 Torino, Italy; r.roberto@studenti.polito.it
[2] Thales Alenia Space, Strada Antica di Collegno 253, 10146 Torino, Italy; andrea.dottavio@thalesaleniaspace.com (A.D.); giorgio.fasano@thalesaleniaspace.com (G.F.)
[3] Department of Management Science & Information Systems, Rutgers Business School—New Brunswick, Rutgers University, Piscataway, NJ 08854, USA; jpinter@business.rutgers.edu
* Correspondence: lorenzo.casalino@polito.it

Abstract

The Lunar Gateway is a planned small space station that will orbit the Moon and serve as a central hub for NASA's Artemis program to return humans to the lunar surface and to prepare for Mars missions. This work presents a hybrid optimization strategy for designing minimum-fuel transfers from an Earth orbit to a Lunar Near-Rectilinear Halo Orbit. The corresponding optimal control problem—crucial for missions to NASA's Lunar Gateway—is characterized by a high-dimensional, non-convex solution space due to the multi-body gravitational environment. To tackle this challenge, a two-stage hybrid optimization scheme is employed. The first stage uses a Genetic Algorithm heuristic as a global search strategy, to identify promising feasible trajectory solutions. Subsequently, the initial solution guess (or guesses) produced by GA are improved by a local optimizer based on a Sequential Quadratic Programming method: from a suitable initial guess, SQP rapidly converges to a high-precision feasible solution. The proposed methodology is applied to a representative cargo mission case study, demonstrating its efficiency. Our numerical results confirm that the hybrid optimization strategy can reliably generate mission-grade quality trajectories that satisfy stringent constraints while minimizing propellant consumption. Our analysis validates the combined GA-SQP optimization approach as a robust and efficient tool for space mission design in the cislunar environment.

Keywords: hybrid optimization; Near Rectilinear Halo Orbit (NRHO); Weak Stability Boundary (WSB); cislunar trajectory design; Lunar Gateway

Check for updates

Academic Editor: Boting Yang

Received: 28 November 2025
Revised: 4 January 2026
Accepted: 13 January 2026
Published: 17 January 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Recently, the European Space Agency has embarked on the Low Earth Orbit (LEO) Cargo Return Services (LCRS) program [1], an initiative designed to develop autonomous European capabilities for cislunar operations utilizing Ariane-6 launches from Kourou Spaceport (French Guiana). Despite significant background in spacecraft systems and orbital mechanics, comprehensive end-to-end mission design studies—validating whether European systems can execute autonomous Earth-to-Gateway transfers using exclusively European infrastructure—have been limited so far. Existing analyses focus primarily on separated trajectory segments or on spacecraft subsystem performance, lacking integrated assessments of complete mission scenarios that simultaneously account for launch vehicle constraints, multi-body orbital mechanics,

propulsion limitations, and operational requirements. The research presented here directly addresses this critical gap by performing a systematic analysis of complete Earth-to-Gateway transfer architectures.

The current circumstances imply a narrow decision window. U.S. commercial providers offer rapidly maturing cislunar transportation capabilities that could establish market dominance before European systems reach operational status. Chinese autonomous cargo operations continue progression toward independent lunar infrastructure. Europe, however, has so far remained absent from this operational landscape. While long-term strategic initiatives such as LCRS and Argonaut (Europe's lunar lander program) exist, no European missions are forecast to achieve operations in the cislunar environment for the next five to eight years, risking that Europe will belatedly arrive when foundational rules and partnerships for lunar operations have already been established by other actors.

Cislunar trajectory optimization represents one of the most challenging problems in modern astrodynamics, central to enabling sustainable lunar exploration and establishing permanent human presence beyond Low Earth Orbit [2]. The difficulty stems from the fundamental nature of multi-body gravitational dynamics, where spacecraft motion in the Earth–Moon system cannot be described by closed-form analytical solutions. Near Rectilinear Halo Orbits (NRHOs) [3] around the Moon exemplify this challenge: these orbits exist within the chaotic regime of the Circular Restricted Three-Body Problem (CR3BP, consult e.g., [2]), exhibiting extreme sensitivity to initial conditions and perturbations. The resulting trajectory design problem is characterized by highly non-convex solution spaces, multiple local optima, and stringent operational constraints on propellant budget, mission duration, and rendezvous timing. Traditional two-body Keplerian mechanics, which govern most Earth-Orbit operations, provide no meaningful approximation in this domain, necessitating fundamentally different approaches to trajectory design and optimization.

In the present study, the emphasis is placed on our methodological contribution: we propose a comprehensive mission design study employing hybrid optimization techniques (see e.g., [4,5]) to address the Earth-to-NRHO transfer problem [6]. While the techniques developed are broadly applicable to a wide range of cislunar mission scenarios, the European Space Agency's LCRS program serves as the primary motivating case study. This application represents a compelling and strategically relevant challenge, and therefore provides the operational context for demonstrating the feasibility and performance of autonomous European access to NASA's Lunar Gateway [7].

The Lunar Gateway, positioned in a synodic resonant NRHO around the Moon, constitutes a critical infrastructure element in humanity's travels to the Moon. Unlike conventional Low Lunar Orbit stations, the Gateway's NRHO provides continuous Earth communication and enables access to any point on the lunar surface with minimal spacecraft velocity change budget. However, these operational advantages come at the cost of significantly improved Earth–Moon transfer trajectories design. The orbit's highly elliptical geometry, combined with strong gravitational perturbations from Earth, Moon and Sun, creates a dynamic environment that is fundamentally different and more challenging (from a classical orbital mechanics point of view). Rendezvous operations require precise timing, as the orbit's orientation and altitude significantly vary throughout its period, and small trajectory errors can lead to large deviations due to the chaotic nature of the dynamics involved.

Our research directly addresses this gap through systematic mission design analyses for complete Earth-to-Gateway cargo transfer scenarios. This will enable us to investigate

whether European lunar cargo services represent achievable objectives worthy of sustained investment, or they require pragmatic reassessment.

To tackle the related challenging optimization problem, a two-stage hybrid scheme is employed. The first stage uses a genetic algorithm (GA) heuristic for global search, to identify promising and feasible trajectory regions. The second stage refines the initial solution guess (or guesses) produced by the global search, using an efficient local optimizer based on the sequential quadratic programming (SQP) method, since SQP rapidly converges to a high-precision, dynamically feasible solution.

The article is structured as follows. Section 2 details the mathematical formulation of the optimization problem, including the equations of motion and the multi-arc trajectory scheme adopted, and the structure of the associated nonlinear programming (NLP) model. Section 3 describes the hybrid optimization approach, explaining the distinct roles of the GA and SQP stages, as well as their integration. Section 4 presents the numerical results for a cargo mission case study, demonstrating the efficiency of the approach adopted. Section 5 provides concluding remarks and discusses potential avenues for future enhancements.

2. Problem Statement

At its core, the problem of designing a spacecraft trajectory is an issue of optimal control, see e.g., [8,9]. The spacecraft's movement is determined by a set of complicated physical laws—the differential equations of motion—which account for the gravity of the Earth, Moon, and Sun, while the spacecraft's position, velocity, and mass define its state at each time moment. The fundamental challenge is to find the optimal control option to guide the spacecraft from its starting point to its final destination while minimizing a chosen performance index, such as the total amount of fuel consumed. The trajectory must satisfy a set of crucial mission constraints, like arriving at the correct orbit, not crashing into any celestial bodies, and staying within the physical limits of the spacecraft's engine performance. To solve this problem, we must create a precise mathematical model of the spacecraft's dynamics, define exactly what we want to optimize as the objective function, and clearly lay out all the rules and boundaries that constitute the constraints of the optimization problem.

State vector and equations of motion. The state vector, denoted by $\mathbf{s}(t)$, is the set of variables that uniquely describes the dynamics condition of the vehicle at any time t. The equations of motion are expressed in an Earth-Centered Inertial (ECI) reference frame, which is aligned with the J2000 epoch (a standard celestial reference frame in astronomy, see [10]). The system state $\mathbf{s}(t)$ is represented by a seven-dimensional vector including position $\mathbf{r}(t)$, velocity $\mathbf{v}(t)$, and mass m(t):

$$\mathbf{s}(t) = \begin{bmatrix} \mathbf{r}(t) \\ \mathbf{v}(t) \\ m(t) \end{bmatrix}, \tag{1}$$

where $\mathbf{r}(t)$ and $\mathbf{v}(t)$ are the ECI position and velocity vectors. The spacecraft's acceleration, $\mathbf{a}(t)$, is decomposed into gravitational, propulsive, and disturbance terms:

$$\mathbf{a}(t) = \frac{d\mathbf{v}}{dt} = \mathbf{a}_g(\mathbf{r}, t) + \mathbf{a}_T(\mathbf{u}, t) + \mathbf{a}_d(\mathbf{r}, \mathbf{v}, t). \tag{2}$$

Here, $\mathbf{a}_T$ is the non-conservative propulsive acceleration, which acts as the control variable. Other non-conservative perturbations, such as solar radiation pressure or atmospheric drag, which would be included in the $\mathbf{a}_d(\mathbf{r}, \mathbf{v}, t)$ term in Equation (2), have limited relevance for Earth–NRHO transfers and are neglected.

The dominant term, $\mathbf{a}_g$, represents the combined gravitational force from all celestial bodies included in the model. Instead of a simple two-body problem (e.g., just Earth and the spacecraft), this term represents an n-body problem, summing the gravitational forces exerted by the Earth, Moon, and Sun simultaneously. This high-fidelity model is essential for accurately capturing intricate dynamics of cislunar space. The gravitational acceleration is given by:

$$\mathbf{a}_g(\mathbf{r}, t) = -\frac{\mu_{\text{Earth}}\mathbf{r}_{sc}}{|\mathbf{r}_{sc}|^3} + \sum_{j=2}^{n} \mu_j \left(\frac{\mathbf{r}_j - \mathbf{r}_{sc}}{|\mathbf{r}_j - \mathbf{r}_{sc}|^3} - \frac{\mathbf{r}_j}{|\mathbf{r}_j|^3} \right) \tag{3}$$

Here, the gravitational acceleration $\mathbf{a}_g(r, t)$ is composed of the primary two-body attraction from Earth and the perturbing effects of other celestial bodies. The first term, $-\frac{\mu_{\text{Earth}}\mathbf{r}_{sc}}{|\mathbf{r}_{sc}|^3}$, represents the acceleration due to Earth's gravity, where μ_{Earth} is Earth's gravitational parameter and r_{sc} is the position vector of the spacecraft in the Earth-Centered Inertial frame (see e.g., [2]). The summation $\sum_{j=2}^{n}$, accounts for the gravitational influence of n third-body perturbers (such as the Sun and Moon). For each perturbing body j, μ_j is its gravitational parameter and $\mathbf{r}_j$ is its position vector in the ECI frame. The expression within the parentheses consists of two components: the direct term, $\frac{\mathbf{r}_j - \mathbf{r}_{sc}}{|\mathbf{r}_j - \mathbf{r}_{sc}|^3}$, which is the gravitational acceleration exerted by body j directly on the spacecraft, and the indirect term, $-\frac{\mathbf{r}_j}{|\mathbf{r}_j|^3}$, which corrects for the non-inertial acceleration of the Earth due to that same body. Finally, the change in spacecraft mass is exclusively due to propellant consumption. This is governed by the thruster's performance, specifically its thrust magnitude $T(t)$ and specific impulse I_{sp}:

$$\frac{dm(t)}{dt} = -\frac{T(t)}{I_{sp} g_0}, \tag{4}$$

where g_0 is the standard gravitational acceleration at sea level. This equation links the vehicle dynamics directly to the efficiency of the propulsion system, as a higher I_{sp} results in a lower mass flow rate for a given thrust level, thus saving propellant.

Multiple trajectory arcs and NLP problem formulation. To make the continuous optimal control problem solvable with numerical methods, the trajectory is subdivided (discretized) into a sequence of distinct segments. In our formulation, we assume that the entire trajectory is partitioned into a sequence of ballistic arcs, where the spacecraft simply coasts under the influence of gravity. An impulsive maneuver is executed at the initial time of each arc, denoted as an impulsive $\Delta\mathbf{v}_i$, is applied to represent a propulsive burn. Between these maneuvers, the spacecraft's path is governed solely by then n-body dynamics.

This approach effectively transforms the problem from one with a continuous control function (the engine thrust over time) into a finite-dimensional nonlinear programming (NLP) problem. To adhere to the practical limits of a real propulsion system, the total number of maneuvers is fixed at a maximum of N, where N represents the number of connection points, or nodes, along the trajectory. The value N = 6 is selected in our study, considering that at most 5 maneuvers can effectively attain the required transfer.

The problem is now defined by a finite set of decision variables to be determined. These are:

- The three components of the impulsive velocity change $\Delta\mathbf{v_i} \in \mathbb{R}^3$, applied at each of the N maneuver points.
- The duration τ_j, of each of the $N-1$ ballistic arcs, expressed as a non-dimensional fraction of an assigned maximum time T.

By structuring the problem this way, the trajectory can be constructed segment by segment. For any given set of decision variables, $\mathbf{w} = \left[\Delta\mathbf{v}1, \ldots, \Delta\mathbf{v}_N, \tau_1, \ldots, \tau_{N-1}\right]^\top$, the full path is generated by sequentially applying each $\Delta\mathbf{v}_i$ and then integrating the equations of motion for the duration t_j. This ensures that the state of the spacecraft is continuous from one arc to the next. The problem is now reduced to finding the optimal set of variables $\mathbf{w}$ that minimizes an objective function $J(\mathbf{w})$ while satisfying a set of inequality constraints $\mathbf{g}(\mathbf{w}) \leq \mathbf{0}$ and staying within the allowed bounds on the decision variables themselves. So the finite-dimensional NLP can be stated in its general form as:

$$\begin{aligned} \min_{\mathbf{w}} \quad & J(\mathbf{w}) \\ \text{s.t.} \quad & \mathbf{g}(\mathbf{w}) \leq \mathbf{0}, \\ & \mathbf{W_L} \leq \mathbf{w} \leq \mathbf{W_U}, \end{aligned} \tag{5}$$

where $\mathbf{w}$ represents the vector of decision variables, bounded by the lower and upper bounds $\mathbf{W_L}$ and $\mathbf{W_U}$, respectively. The function components of $\mathbf{g}(\mathbf{w})$, are evaluated through the integration of the equations of motion (1)–(4).

Optimization objective and constraints. The specific optimization criterion is to minimize the propellant consumption. For the impulsive maneuvers considered here, this is physically equivalent to minimizing the total required velocity change, denoted by $\Delta\mathbf{v}$. The objective function, $J(\mathbf{w})$, to be minimized is therefore the sum of the magnitudes of all the impulsive velocity changes ($\Delta\mathbf{v}_i$):

$$J(\mathbf{w}) = \sum_{i=1}^{N-1} \|\Delta\mathbf{v}_i\|. \tag{6}$$

This objective is subject to a set of path and terminal constraints, collected in the vector $\mathbf{g}(\mathbf{w})$, which ensure the final trajectory meets all mission requirements. For the case study considered, these are expressed as a set of normalized inequalities:

$$\mathbf{g}(\mathbf{w}) = \begin{bmatrix} (h(\mathbf{w}) - H_{\max})/S_h \\ (H_{\min} - h(\mathbf{w}))/S_h \\ (i(\mathbf{w}) - I_{\max})/S_i \\ (I_{\min} - i(\mathbf{w}))/S_i \\ (r_{\text{apo}}(\mathbf{w}) - R_{\text{apo,max}})/S_r \\ (R_{\text{apo,min}} - r_{\text{apo}}(\mathbf{w}))/S_r \\ C_{3\min} - C_3(\mathbf{w}) \\ C_3(\mathbf{w}) - C_{3\max} \end{bmatrix} \leq \mathbf{0}, \tag{7}$$

where:

- $h(\mathbf{w})$ is the minimum geocentric altitude along the trajectory, constrained to $[H_{\min}, H_{\max}]$.
- $i(\mathbf{w})$ is the final orbital inclination at arrival, constrained to $[I_{\min}, I_{\max}]$.
- $r_{\text{apo}}(\mathbf{w})$ is the maximum geocentric distance (apogee) of the transfer, constrained to $[R_{\text{apo,min}}, R_{\text{apo,max}}]$ to ensure the correct path is followed.
- $C_3(\mathbf{w})$ is the characteristic energy at the final perigee, constrained to $[C_{3,\min}, C_{3,\max}]$ for a low-energy arrival. (see e.g., [2])
- S_h, S_i, S_r are scaling factors used to normalize the constraints and improve the numerical conditioning of the NLP problem.

3. Optimization Approach

The solution of the nonlinear programming problem, formulated in Section 2, is approached by iterative optimization algorithms that require repeated evaluations of the objective and constraint functions. In this work, the $\mathbf{g}(\mathbf{w})$ functions are evaluated using a "black box": a trajectory propagator that encapsulates the high-fidelity n-body equations of motion. The NLP solver provides this propagator with a "candidate" decision vector $\mathbf{w} = [\Delta\mathbf{v}_1, \ldots, \Delta\mathbf{v}_{N-1}, \tau_1, \ldots, \tau_{N-1}]^T$. The optimization process then necessitates the evaluation of the full trajectory. This is accomplished by the propagator executing a backward integration loop over the $N-1$ segments, starting from the final arc and proceeding to determine the overall initial state. Given the pronounced dynamical instability inherent to NRHO within the multi-body environment, which makes forward integration numerically ill-conditioned, the trajectory is judiciously initiated via backward propagation from a selected arrival state on the NRHO, a strategy that allows for the identification and evaluation of different optimal solutions based on the chosen destination. The resulting state and the full computed trajectory are then returned to the NLP solver to calculate the objective cost and constraint values for the specific input vector of decision variables $\mathbf{w}$.

The underlying multi-body dynamics render the solution space strongly nonconvex. This scenario poses significant numerical challenges for the NLP solver. Challenges include searching for a feasible solution, the existence of multiple local minima, and high sensitivity to the initial guess. Consequently, relying solely on traditional local optimization methods would not be sufficient: they may not find feasible solutions or could converge to a locally optimal solution and miss the global solution. Therefore, a two-stage hybrid methodology is adopted to solve the NLP problem (5). This overall strategy combines the strengths of global and local optimization algorithms aiming to drive the search for the global optimal decision vector:

- First, a Genetic Algorithm (GA) is employed to robustly explore the nonconvex solution space. The GA through the black-box propagator identifies promising regions containing good guess solutions (see e.g., [11–14]).
- Next, a sequential quadratic programming (SQP) method is used to refine the best solution found by the GA. Leveraging this strong initial guess, SQP provides rapid and precise convergence to the local optimum (consult e.g., [15]).

Using the same black-box propagator for both the GA and SQP stages ensures that feasibility and cost are evaluated with an identical numerical model, guaranteeing consistency throughout the entire optimization process.

Penalty functions and constraint handling in GA. The genetic algorithm stage employs a penalty function approach to handle the complicated constraint structure of the problem. Rather than enforcing hard constraints, which could lead to widespread infeasibility within the population, a penalty scheme guides the evolutionary search toward constraint-compliant regions while preserving the needed "genetic diversity" for global scope search.

The fitness function $F(\mathbf{w})$ is therefore formulated as a weighted sum of the cost objective and multiple penalty terms:

$$F(\mathbf{w}) = WJ(\mathbf{w}) + \sum_{k=1}^{N_p} W_k P_k(\mathbf{w}). \tag{8}$$

Here, $J(\mathbf{w})$ is the original objective function, W the associated weight, and the summation is intended over a total of N_p penalty functions. Each term $P_k(\mathbf{w})$ represents a specific penalty designed to enforce a mission constraint or a desired trajectory feature, while its corresponding weight W_k is adapted as needed to balance its relative importance. The specific penalties, indexed by k, are reported here below.

$k = 1$ *Apogee corridor.* A quadratic penalty is applied if the trajectory's maximum geocentric distance, r_{apo}, falls outside the target range $[R_{\text{apo,min}}, R_{\text{apo,max}}]$. This is crucial for achieving the necessary solar perturbations.

$k = 2$ *Apogee quadrant.* A large constant penalty is applied if the apogee occurs in the wrong quadrant relative to the Sun-Earth line (i.e., if the dot product between the Sun's position vector and the spacecraft's apogee position vector is positive), ensuring the correct geometry for a WSB transfer (see e.g., [16]).

$k = 3$ *Characteristic energy.* A quadratic penalty is applied if the departure energy C_3 at the final perigee is outside the narrow range $[C_{3min}, C_{3\,\text{max}}]$, ensuring a low-energy arrival.

$k = 4$ *Altitude at arrival.* A quadratic penalty enforces the final minimum altitude to be within the target range $[H_{\text{min,target}}, H_{\text{max,target}}]$.

$k = 5$ *Inclination at arrival.* A quadratic penalty ensures the final inclination i falls within the target NRHO corridor $[I_{\text{min,target}}, I_{\text{max,target}}]$.

$k = 6$ *Monotonic escape.* A large constant penalty is applied if the spacecraft, after first exiting a predefined radius around the Moon, re-enters it. This critical constraint prevents inefficient lunar flybys.

$k = 7$ *Safety.* A large constant penalty is added if the final minimum altitude drops below a critical safety threshold (e.g., 100 km), preventing atmospheric re-entry.

Penalties 1–3 (WSB trajectory shaping) guide the optimizer toward the characteristic high-energy, high-apogee WSB corridor [6]. Penalties 4–7 (arrival and safety) ensure the trajectory meets its target conditions and operational limits. The choice of the weights is done after a few trial GA runs to balance the constraints influence; the specific values assigned to the weights (w_i) for the shaping objectives and the large penalty constants used for hard constraints are detailed in Table 1. The candidate solutions identified by the GA are subsequently refined using an SQP algorithm.

Table 1. Optimization weights, scale factors, and penalty types used in the fitness function.

Objective/Constraint	**Weight/Penalty (w_i)**	**Scale Factor/Threshold**
Optimization Objectives (Soft):		
Propellant Efficiency (Δv)	1000	Δv_{max}
Target Inclination ($k = 5$)	400	5°
Target Perigee Altitude ($k = 4$)	100	100 km
Apogee Corridor ($k = 1$)	10	10^5 km
Characteristic Energy ($k = 3$)	10	0.5 km^2/s^2
Hard Constraints (Safety/Geometry):		
Apogee Quadrant ($k = 2$)	10^{12}	Dot Product > 0
Monotonic Escape ($k = 6$)	10^{10}	Radius Re-entry
Safety Altitude ($k = 7$)	10^9	$<$100 km

The NLP solved during this local refinement stage is based on the mathematical model expressed by relations (5)–(7).

GA pseudocode and SQP pseudocode. For completeness, pseudocodes for general Algorithms 1 and 2 are included below (see e.g., [15,17]). Comments added to the pseudocodes are denoted by ▷.

Algorithm 1 Pseudocode for a generic Genetic Algorithm (GA)

Given:
  Fitness function $f(x)$ to MINIMIZE
  Decision encoding (real-valued, binary, etc.)
  Population size N
  Crossover rate p_c, mutation rate p_m
  Tournament size k (for selection)
  Elitism count E (usually 1–5% of N)
  Max generations $G_{\max}$ (or other stopping test)
  Penalty function $P(x) \geq 0$ for constraint violation (optional)

procedure GENETIC_ALGORITHM
  $gen \leftarrow 0$
  $POP \leftarrow$ InitializePopulation(N) ▷ Random within bounds
  for each $x \in POP$ **do**
    x.fit $\leftarrow$ FitnessWithPenalty(x) ▷ $f(x) + P(x)$; lower is better
  end for
  $BEST \leftarrow \text{argmin}_{x \in POP}\, x$.fit
  while not StoppingCondition$(gen, BEST, G_{\max})$ **do**
    ▷ ---- Elitism ----
    $ELITES \leftarrow$ TopEIndividuals(POP, E)
    ▷ ---- Create offspring ----
    $OFFSPRING \leftarrow \emptyset$
    while $|OFFSPRING| < N - E$ **do**
      $p1 \leftarrow$ TournamentSelect(POP, k)
      $p2 \leftarrow$ TournamentSelect(POP, k)
      if rand$() < p_c$ **then**
        $(c1, c2) \leftarrow$ Crossover$(p1, p2)$
      else
        $c1 \leftarrow$ Clone$(p1)$; $c2 \leftarrow$ Clone$(p2)$
      end if
      $c1 \leftarrow$ Mutate$(c1, p_m)$
      $c2 \leftarrow$ Mutate$(c2, p_m)$
      $c1 \leftarrow$ RepairOrProject$(c1)$
      $c2 \leftarrow$ RepairOrProject$(c2)$
      $OFFSPRING \leftarrow OFFSPRING \cup \{c1, c2\}$
    end while
    $OFFSPRING \leftarrow$ TruncateToSize$(OFFSPRING, N - E)$
    ▷ ---- Evaluate ----
    for each $y \in OFFSPRING$ **do**
      y.fit $\leftarrow$ FitnessWithPenalty(y)
    end for
    ▷ ---- Next generation ----
    $POP \leftarrow ELITES \cup OFFSPRING$
    $gen \leftarrow gen + 1$
    if $\min_{y \in OFFSPRING} y$.fit $< BEST$.fit **then**
      $BEST \leftarrow \text{argmin}_{y \in POP}\, y$.fit
    end if
  end while
  return $BEST$
end procedure

function FITNESSWITHPENALTY(x)
  return $f(x) + P(x)$
end function

function STOPPINGCONDITION(gen, BEST, G_max)
  return $(gen \geq G_{\max})$ OR (NoImprovement) OR $(BEST$.fit $\leq$ target)
end function

https://doi.org/10.3390/a19010080

Algorithm 2 Pseudocode for a generic Sequential Quadratic Programming (SQP) algorithm

Given:
  x_0 ▷ Initial solution guess
  $\lambda_{e0}, \lambda_{i0} \geq 0$ ▷ Initial multipliers (optional; else zeros)
  $B_0 \succ 0$ ▷ Initial Hessian approximation (e.g., γI)
  Tolerances: $\varepsilon_{\text{stat}}, \varepsilon_{\text{feas}}, \varepsilon_{\text{comp}}$
  Line-search parameters: $\alpha_0 \in (0,1], \rho \in (0, 1/2), \beta \in (0,1), \alpha_{\min} > 0$
  Penalty parameters: $\sigma_f > 0, \sigma_c > 0$ ▷ For exact-penalty merit
  Max iterations: $K_{\max}$

procedure SQP_ALGORITHM
  $k \leftarrow 0$; $x \leftarrow x_0$; $\lambda_e \leftarrow \lambda_{e0}$; $\lambda_i \leftarrow \lambda_{i0}$; $B \leftarrow B_0$
  repeat
    ▷ ---- Evaluate model at x ----
    $f \leftarrow f(x)$; $g \leftarrow \nabla f(x)$
    $c_e \leftarrow c_e(x)$; $J_e \leftarrow J_e(x)$ ▷ Jacobian of equalities
    $c_i \leftarrow c_i(x)$; $J_i \leftarrow J_i(x)$ ▷ Jacobian of inequalities
    ▷ ---- Check KKT-like stopping conditions ----
    stationarity_resid $\leftarrow \|g + J_e^T \lambda_e + J_i^T \lambda_i\|_\infty$
    feas_resid $\leftarrow \max(\|c_e\|_\infty, \|\max(0, c_i)\|_\infty)$
    comp_resid $\leftarrow \|\lambda_i \odot \max(0, c_i)\|_\infty$
    if stationarity_resid $\leq \varepsilon_{\text{stat}}$ **and** feas_resid $\leq \varepsilon_{\text{feas}}$ **and** comp_resid $\leq \varepsilon_{\text{comp}}$ **then**
      return $(x, \lambda_e, \lambda_i)$
    end if
    ▷ ---- Form and solve QP subproblem ----
    ▷ Quadratic model of Lagrangian with linearized constraints:
    $\min_d \quad \frac{1}{2} d^T B d + g^T d$
    s.t. $\quad J_e d + c_e = 0$
    $J_i d + c_i \leq 0$
    $(d, \Delta\lambda_{e,\text{QP}}, \Delta\lambda_{i,\text{QP}}) \leftarrow \text{SolveQP}(B, g, J_e, c_e, J_i, c_i)$
    ▷ ---- Line search on exact-penalty merit function ----
    $\alpha \leftarrow \alpha_0$
    while $\phi(x + \alpha d; \sigma_f, \sigma_c) > \phi(x; \sigma_f, \sigma_c) + \rho\alpha\nabla\phi(x)^T d$ **do**
      $\alpha \leftarrow \beta\alpha$
      if $\alpha < \alpha_{\min}$ **then** ▷ Check minimum step tolerance
        $d \leftarrow \text{RestorationDirection}(x)$ ▷ Optional fallback to reduce infeasibility
        $\alpha \leftarrow \alpha_0$
        break
      end if
    end while
    ▷ ---- Take step ----
    $x^+ \leftarrow x + \alpha d$
    ▷ ---- Update multipliers ----
    $\lambda_e^+ \leftarrow (1-\theta)\lambda_e + \theta(\lambda_e + \Delta\lambda_{e,\text{QP}})$ ▷ $\theta \in (0,1]$, often $\theta = 1$
    $\lambda_i^+ \leftarrow \text{proj}_{\geq 0}((1-\theta)\lambda_i + \theta(\lambda_i + \Delta\lambda_{i,\text{QP}}))$
    ▷ ---- BFGS (Powell-damped) update on Lagrangian Hessian approx ----
    $s \leftarrow x^+ - x$
    $g_L \leftarrow g + J_e^T \lambda_e + J_i^T \lambda_i$
    $g_L^+ \leftarrow \nabla f(x^+) + J_e(x^+)^T \lambda_e^+ + J_i(x^+)^T \lambda_i^+$
    $y \leftarrow g_L^+ - g_L$
    if $s^T y \geq 0.2 s^T B s$ **then**
      $\tilde{y} \leftarrow y$
    else
      $\tau \leftarrow \frac{0.8 s^T B s}{s^T B s - s^T y}$ ▷ Powell damping factor
      $\tilde{y} \leftarrow \tau y + (1-\tau) B s$
    end if
    if $s^T \tilde{y} > 10^{-12}$ **then**
      $B \leftarrow B - \frac{B s s^T B}{s^T B s} + \frac{\tilde{y}\tilde{y}^T}{s^T \tilde{y}}$
    end if
    ▷ ---- Next iterate ----
    $x \leftarrow x^+$; $\lambda_e \leftarrow \lambda_e^+$; $\lambda_i \leftarrow \lambda_i^+$
    $k \leftarrow k + 1$
  until $k \geq K_{\max}$
  return $(x, \lambda_e, \lambda_i)$ ▷ Best found; report termination reason
end procedure

4. Numerical Implementation and Results

The optimization process is built around a shared computational framework used by both the global and local stages. Trajectory propagation is handled by the high-fidelity

n-body model detailed in Equation (3), which incorporates the gravitational effects of Earth, Moon, and Sun. The dynamics equations are integrated using the following key implementation features:

- *Integration scheme.* MATLAB's ode45 solver with RelTol = 10^{-11} and AbsTol = 10^{-12}, ensuring numerical accuracy over extended 50–150 day arcs.The propagation follows a segmented approach, where each arc is integrated separately. Specifically, the final state of the i-th arc defines the initial position for the subsequent segment, while the velocity is updated by the impulsive maneuver $\Delta \mathbf{v}_i$ before the next integration call. For details, consult https://www.mathworks.com/help/matlab/ref/ode45.html (accessed on 20 December 2025).
- *Reference frame.* All states and ephemerides are expressed in the Earth–Centered Inertial J2000 frame (also referred to as EME2000 in SPICE, [10]), with origin at Earth's center of mass and axes fixed with respect to the mean equator and equinox at epoch J2000. The X–axis points toward the mean vernal equinox of J2000, the Z–axis aligns with Earth's mean rotation axis at J2000, and the Y–axis completes a right–handed triad in the equatorial plane. This realization is quasi–inertial and consistent with the ICRF at the J2000 epoch to first order, making it appropriate for high–precision orbital propagation over mission timescales considered here. SPICE kernels are queried explicitly in the 'J2000' inertial frame, and no transformations to rotating Earth–fixed frames are applied during numerical integration. Figure 1 provides a visual representation of the ECI J2000 frame.
- *Time normalization.* Time normalization is a numerical technique that improves solver performance by re-scaling the optimization variables. Instead of working with physical arc durations Δt_i in seconds, the optimizer manipulates a dimensionless variable τ_i, where the two are related by a characteristic time constant T via $\Delta t_i = \tau_i \cdot T$.

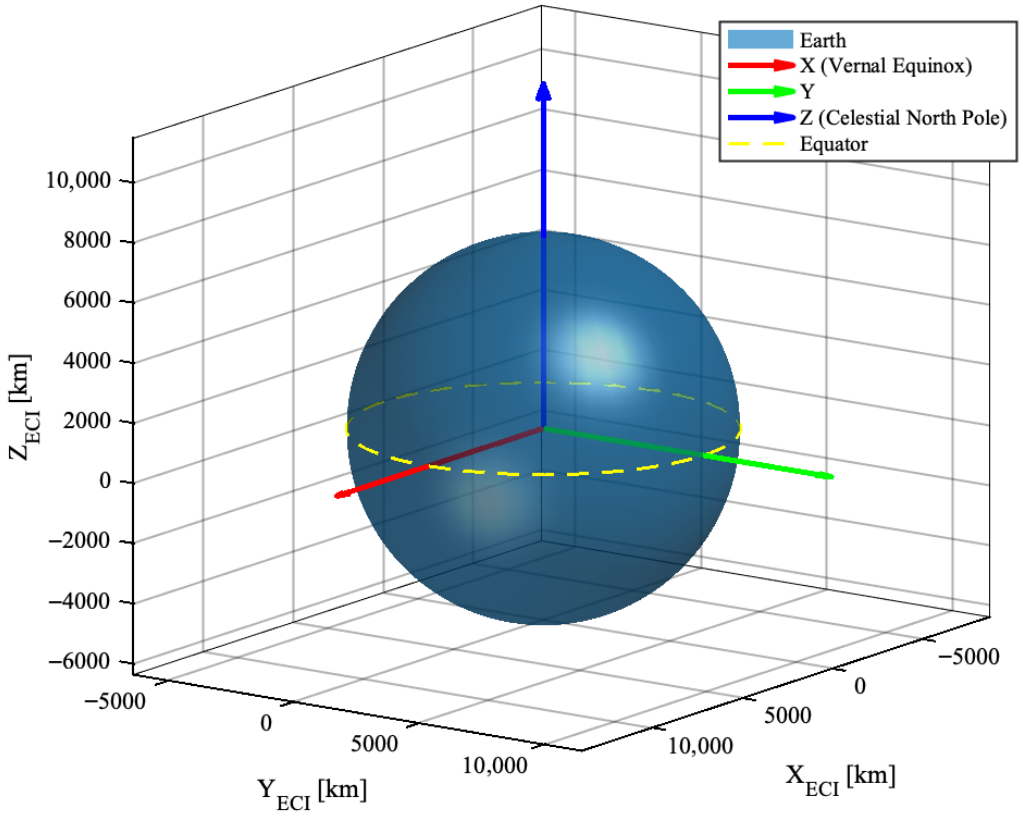

Figure 1. ECI reference frame.

This model captures third-body perturbations and provides a robust foundation for both global GA sampling and local SQP refinement, ensuring that optimized trajectories remain valid under realistic dynamical conditions.

4.1. Case Study: A Cargo Mission

This case study investigates a low-energy transfer from a Low Earth Orbit to a Near-Rectilinear Halo Orbit. The scenario is modeled on the CAPSTONE mission profile [6] and applies the characteristics of an uncrewed cargo vehicle. The trajectory leverages a weak-stability-boundary (WSB) transfer to minimize propulsive cost over a mission duration of 120–150 days. The entire transfer is partitioned into multiple arcs, corresponding to $N = 6$ nodes. $N - 1 = 5$ admissible impulsive maneuvers are considered; the segment durations are parameterized by normalized variables $\tau_i \in [0.02,\ 1]$ scaled by a 50-day reference to promote well-conditioned timings in the optimizer. The decision vector comprises the three components of each $\Delta\mathbf{v}_i$ and τ_i, while the optimization criterion is the minimization of the total impulse defined by (6).

The constraint set encodes feasibility and WSB geometry: minimum geocentric altitude in the corridor $h_{\min} \in [170,\ 250]$ km, arrival inclination $i \in [5^\circ,\ 7^\circ]$, Earth-centered apogee distance $r_{\text{apo}} \in [1.1 \times 10^6,\ 1.8 \times 10^6]$ km, and characteristic energy at departure $C_3 \in [-0.7,\ -0.3]$ km^2/s^2. Additional safety margins enforce a minimum Earth–Moon clearance along the arc and exclude hyperbolic Earth escape at perigee. These bounds are applied in normalized form within the NLP to balance constraint scales and improve convergence.

Global exploration uses a genetic algorithm to sample the highly nonconvex landscape and identify near-feasible guesses that exhibit high-apogee shaping and soft lunar approach consistent with the WSB capture. The GA fitness augments the total Δv with smooth penalties for constraint deviations, discouraging segment collapse and favoring admissible apogee and C_3 windows. Local refinement adopts SQP with analytic first derivatives; feasibility is driven below prescribed tolerances while redistributing maneuver effort and segment durations to improve the Δv.

Representative solutions produced by the GA $\rightarrow$ SQP process exhibit:

i An apogee near 1.3×10^6 km, where solar perturbations shape the inertial ellipse;
ii A gradual Earth departure satisfying the C_3 range;
iii A low-energy lunar arrival that meets the target inclination corridor;
iv The optimal thrust profile is characterized by a sequence of major propulsive burns concentrated at the initial perigee and on the pre-apogee segment of the trajectory, concluding with low-magnitude maneuvers for terminal phase correction.

The refined trajectories maintain altitude and Moon-proximity margins, while reducing the total Δv relative to GA guesses, with maneuver count and placement consistent with the multiple-shooting design.

Vehicle-level constraints. In addition to the geometric and trajectory feasibility constraints, the NLP formulation incorporates a set of vehicle-specific performance limits derived from the spacecraft platform used for the cargo mission [1]. These bounds reflect the system's operational capabilities, such as its total mass budget and propulsion performance. These limits are implemented in the optimization problem as normalized inequality constraints (the relevant numerical values are strictly confidential).

- *Mass budget and propellant usage.* The initial mass of the spacecraft, m_0, is the sum of its dry mass (m_{dry}), propellant mass (m_p), and payload mass (m_{pl}). The usable propellant is bounded by $0 \leq m_p \leq m_{p,\max}$, and the final mass, m_f, must be greater than or equal to the dry mass plus the payload mass. Propellant consumption for each maneuver is linked to the velocity change, Δv_i, via Tsiolkovsky's rocket equation:

$\|\Delta \mathbf{v}_i\| \leq v_e \ln\left(m_i^+ / m_i^-\right)$, where $v_e = I_{sp} g_0$ is the effective exhaust velocity (a function of the specific impulse I_{sp} and standard gravity g_0), and m_i^+ and m_i^- are the spacecraft masses just before and after the i-th burn, respectively.

- *Thrust and acceleration limits.* The thrust magnitude, T_i, produced during each burn must be within the engine's operational range, i.e., $T_{\min} \leq T_i \leq T_{\max}$. This, in turn, limits the acceleration that the spacecraft can achieve, as the thrust acceleration magnitude, $\|\mathbf{a}_{T,i}\| = T_i / m$, must not exceed a maximum value, $a_{\max}$.
- *Operational constraints.* Several practical mission rules are enforced. These include a maximum number of engine burns (at most N), a minimum time interval between consecutive burns, and bounds on the duration of each coasting segment, $\tau_{\min} \leq \tau_i \leq \tau_{\max}$. Additional constraints can also be placed on the total number of engine restarts, the allowable windows for thrusting (duty cycles).
- *Power and thermal margins.* For the selected propulsion technology, the steady-state thrust and the cumulative burn time must respect the available power from the solar panels and the thermal limits of the engine system to prevent overheating.

These vehicle-level constraints are applied in normalized form within $\mathbf{g}(\mathbf{w})$ and are active in both stages: as penalty terms in the GA composite fitness and as explicit constraints in the SQP refinement.

Our case study yields several key insights into the optimization strategy. First, enforcing constraints on the apogee altitude and departure energy (C_3) effectively guides the search towards low-energy corridors, enabling significant propellant savings at the cost of an extended transfer duration.

Finally, the analysis reveals a sensitivity to both the departure epoch and the insertion point in the NRHO. This suggests that a purely local optimization from a single guess may not be sufficient. A more robust approach would involve initiating a series of local optimizations (a multi-start approach) from slightly varied initial conditions centered around a promising GA solution. Furthermore, before finalizing a trajectory, its robustness should be validated by introducing small perturbations to the segment times and maneuver epochs to ensure the solution remains stable and feasible.

4.2. Global Optimization Search

This Section illustrates, at a detailed level, the global optimization process considered. For details, consult https://www.mathworks.com/help/gads/ga.html (accessed on 2 December 2025).

Solver configuration. All solver options were kept at their MATLAB R2024a defaults, except PopulationSize $[P]$ and Generations $[G]$, which were set respectively to achieve convergence within the desired runtime budget (all other operators and tolerances remain at the default values as per MathWorks documentation). It is important to note that the algorithm performs well without the need of tuning its parameters, highlighting the flexibility of the proposed approach. The relevant GA parameters are listed here.

- PopulationSize: 50; Generations: 250;
- CreationFcn (default): *gacreationuniform* or *gacreationlinearfeasible* depending on presence of linear constraints;
- CrossoverFcn (default): *crossoverscattered*; *CrossoverFraction* $= 0.8$;
- MutationFcn (default): *mutationgaussian* (unconstrained) or *mutationadaptfeasible* (constrained);
- SelectionFcn (default): *selectionstochunif*; FitnessScalingFcn (default): *fitscalingrank*;
- Termination (default): FunctionTolerance and MaxStallGenerations criteria; also FitnessLimit if set;
- ConstraintTolerance (default): 10^{-3}.

The implementation was carried out in MATLAB R2024a, utilizing the Global Optimization Toolbox and the Parallel Computing Toolbox. With this setup, the GA runtime is approximately 5 min on a workstation equipped with a 24-core Intel Core i9 processor and 32 GB of RAM.

Constraint satisfaction and physical consistency. The typical evolution of the performance index during a GA run is shown in Figure 2; multiple GA runs provide very similar results and trajectories.The selected GA solution satisfies the mission feasibility constraints (altitude range, arrival inclination, C_3, apogee, safety margins, temporal regularity) within the prescribed normalized tolerances $[\leq 10^{-3}]$, as evaluated along the backward-propagated trajectory. Dynamics compliance is ensured by construction: the same segment-by-segment n-body backward propagator (Earth–Moon–Sun) is used to evaluate the objective and all constraints; propellant usage and mass flow are consistent with thrust/I_{sp} where applicable.

The resulting trajectory is illustrated in Figure 3 and the maneuver timeline (impulse magnitudes and epochs) in Figure 4.

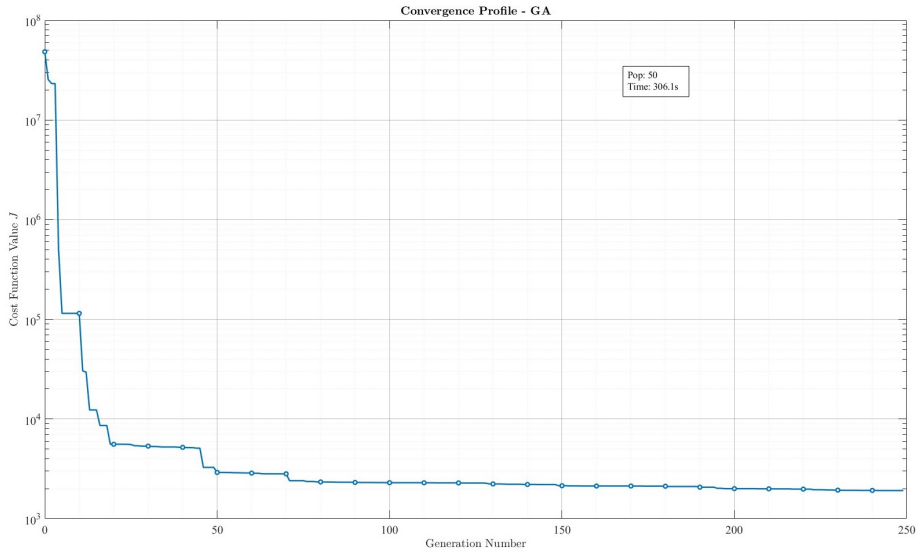

Figure 2. GA convergence—Cost function vs Generation number.

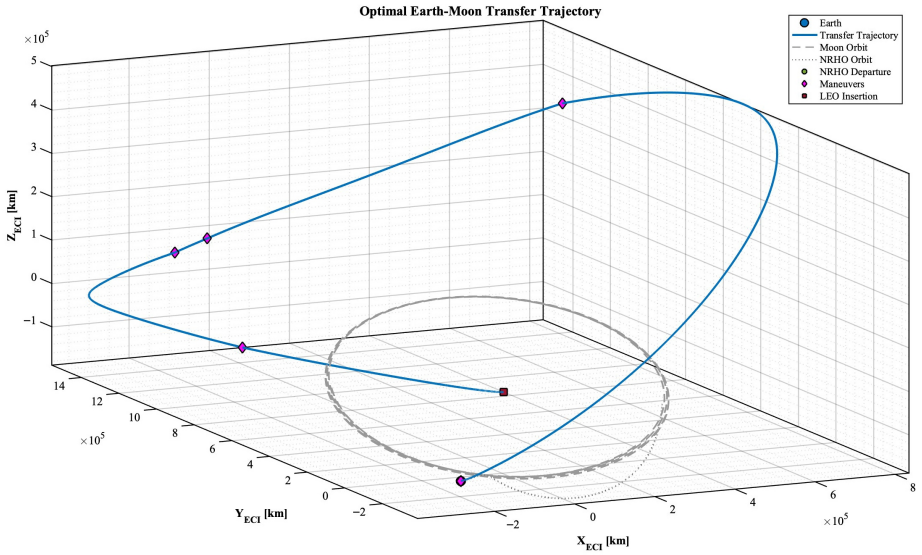

Figure 3. GA solution: propagated trajectory.

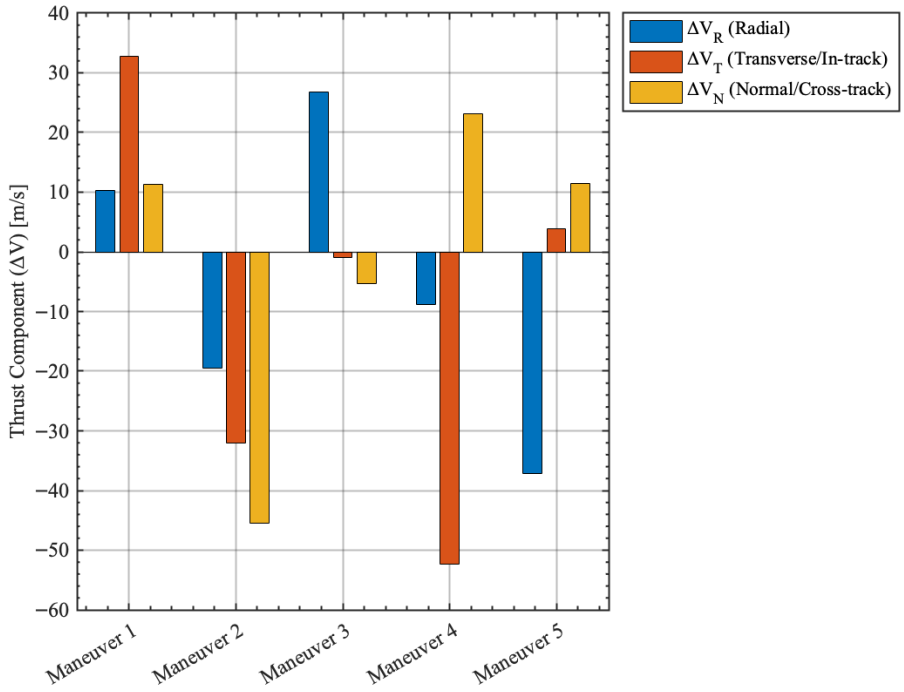

Figure 4. GA solution: maneuver timeline (magnitudes and epochs).

4.3. Local Optimization Search

Solver configuration. The SQP optimizer uses standard settings as reported below. For details, consult https://web.stanford.edu/group/SOL/guides/sndoc7.pdf (accessed on 20 December 2025).

- Warm start: Best GA candidate;
- Gradients: Analytic gradients for $J(\mathbf{w})$ and all constraints; variable/constraint scaling enabled;
- Hessian: Reduced-memory BFGS update of the Lagrangian;
- Line search and merit: Backtracking line search with ℓ_1-merit function and adaptive penalty multipliers;
- Tolerances: Feasibility $\leq [1 \times 10^{-8}]$, optimality $\leq [1 \times 10^{-6}]$; maximum iterations [ItMax]; step-length safeguards $[\varepsilon_{\min}, \varepsilon_{\max}]$;
- Variable bounds: Same box limits on $\Delta\mathbf{v}$ and $\{\tau_j\}$ as in the GA stage;
- Propagation: Identical backward "black-box" n-body propagator per segment ($N-1$ loops), which relies on the MATLAB R2024a *ode45* solver for numerical integration, ensuring model consistency across stages.

The SQP refinement was performed using the SNOPT solver within MATLAB R2024a. The Parallel Computing Toolbox was employed to execute multiple optimizations in parallel (via parfor) starting from the GA candidates. With this setup, the runtime is approximately 1 h on a workstation equipped with a 24-core Intel Core i9 processor and 32 GB of RAM.

Constraint satisfaction and physical consistency. All inequality constraints are satisfied within feasibility tolerance $[\leq 10^{-8}]$ by the refined solution; trajectory metrics computed via the same backward propagation confirm compliance with mission geometry and safety margins. The refined solution achieves an additional reduction in total $\Delta\mathbf{v}$ relative to the GA guess (approximately 77%, decreasing from 219.44 m/s to 51.25 m/s of the SQP solution), demonstrating a substantial enhancement in propellant efficiency compared to the preliminary solution. Furthermore, this final cost aligns with the operational baseline established by the CAPSTONE mission [6], confirming the validity of the obtained trajectory. The refined trajectory is depicted in Figure 5 and the refined maneuver timeline in Figure 6.

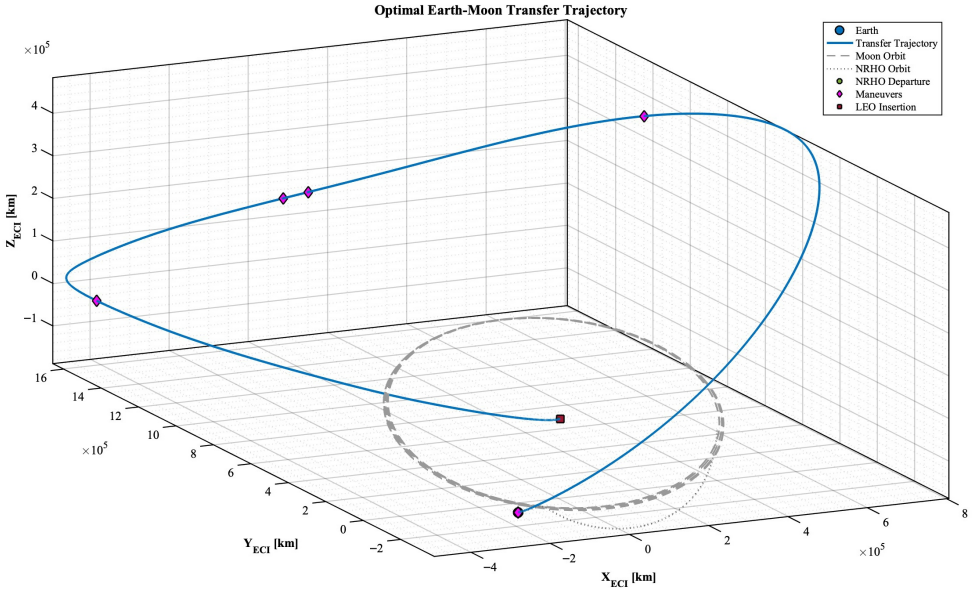

Figure 5. SQP (refined) solution: propagated trajectory.

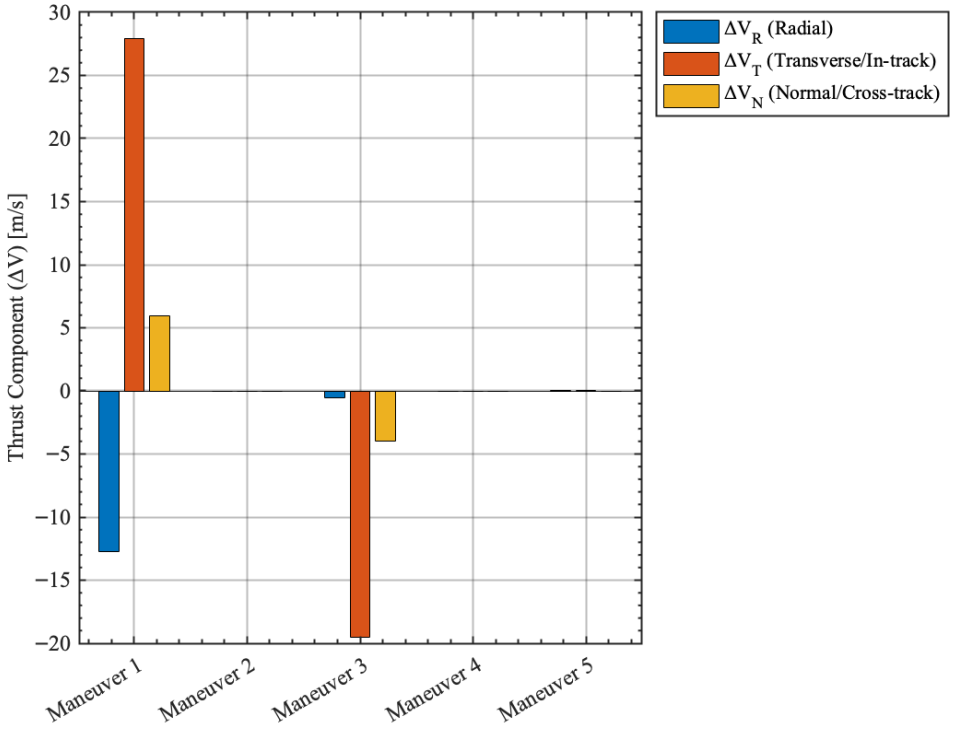

Figure 6. SQP-refined solution: maneuver timeline (magnitudes and epochs).

Efficiency, robustness, and computational limitations. The adopted hybrid GA-SQP strategy constitutes a robust and efficient approach for solving the non-convex optimization problem, capitalizing on the respective strengths of the global and local search paradigms employed. The main feature are listed below.

- *Efficiency.* The strategy's efficiency arises from a structured handover between optimization paradigms. The GA performs the initial (computationally intensive) global exploration, but is terminated once it has successfully identified a promising basin of attraction. This prevents diminishing returns from applying random search as it nears a local optimum. The SQP method then takes over, using the GA's output as a high-quality initial guess. Using analytic gradients and its inherent quadratic convergence

properties, the SQP algorithm performs the final high-precision refinement efficiently, drastically reducing the overall computational cost compared to a standalone global search.

- *Robustness.* The stochastic, population-based nature of the GA provides inherent robustness. By exploring the solution space in parallel from many different starting points, it effectively mitigates the risk of the local optimizer getting trapped in a poor local minimum, which is a significant danger in multi-body problems. The GA's ability to handle non-smooth penalty functions and large, disconnected feasible regions makes it suitable for finding the "right" corridor for the WSB transfer, a task where gradient-based methods would likely fail without a good initial guess.
- *Computational limitations.* Despite its strengths, the computational framework is not without limitations: these aspects are highlighted below.
  - *Parameter Tuning.* The performance of the GA is sensitive to the choice of its parameters (population size, mutation/crossover rates) and the weighting of the penalty functions. Inappropriate tuning can lead to premature convergence or an overly slow exploration of the search space. A degree of problem-specific calibration is required to achieve the best results.
  - *Dimensionality.* The computational cost, particularly for the GA, scales with the dimension of the decision vector (i.e., the number of shooting nodes, N). While parallelization of fitness evaluations can offset this, problems with a high number of nodes may become computationally prohibitive for the global search stage.
  - *SQP Convergence Issues.* While robust, the SQP solution stage can still encounter difficulties. If the initial guess provided by the GA is on the edge of a highly non-linear or ill-conditioned region of the constraint manifold, the local optimizer may struggle to converge or may fail to find a feasible point.

5. Conclusions and Future Work

This paper presents a robust hybrid optimization strategy for designing fuel-efficient, weak-stability-boundary transfers from Earth orbit to a lunar Near Rectilinear Halo Orbit. By combining a Genetic Algorithm for global exploration with a Sequential Quadratic Programming method for local refinement, the proposed framework successfully navigates the nonconvex design space characteristic of multi-body problems. The results of our study demonstrate that the proposed hybrid approach can reliably generate trajectories of "mission-grade" quality. The latter term implies that the resulting solutions are not merely theoretical, but possess the high fidelity and precision required for operational use in an actual space mission. Specifically, the final trajectories simultaneously satisfy all specified constraints—both the geometric requirements of the orbit and the vehicle's own operational limits—to a high degree of numerical accuracy. The ability to consistently produce such robust and precisely converging solutions validates the efficacy of the hybrid GA-SQP scheme for tackling this class of challenging astrodynamical optimization problems.

Future research could extend this framework in several directions. First, the inclusion of non-conservative forces like solar radiation pressure or low-thrust propulsion models, in addition to impulsive maneuvers, would enable the design of even more fuel-efficient trajectories for electric propulsion systems. In addition, this would involve augmenting the optimization problem to account for key uncertainties, such as maneuver execution errors and orbital dynamics dispersions, leading to trajectories with enhanced performance under off-nominal conditions. Finally, applying this methodology to a wider range of cislunar and interplanetary mission scenarios would further demonstrate its versatility as a powerful tool for modern space mission design.

Author Contributions: Methodology, L.C., A.D., G.F., J.D.P. and R.R.; Software, R.R.; Validation, R.R.; Investigation, R.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The orbital mechanics data used in this study are publicly available via the NAIF/JPL SPICE system. Specific vehicle performance data are confidential and cannot be shared.

Conflicts of Interest: Authors A.D. and G.F. are employees of Thales Alenia Space. The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest affecting the results.

References

1. European Space Agency. ESA—European Space Agency. 2025. Available online: https://www.esa.int (accessed on 15 September 2025).
2. Vallado, D.A. *Fundamentals of Astrodynamics and Applications*, 4th ed.; Microcosm Press: Hawthorne, CA, USA, 2013.
3. Zimovan-Spreen, E.M.; Scheuerle, S.T.; Mccarthy, B.P.; Davis, D.C.; Howell, K.C. Baseline orbit generation for near rectilinear halo orbits. In *Proceedings of the AAS/AIAA Astrodynamics Specialist Conference*; NASA Johnson Space Center: Houston, TX, USA, 2023.
4. Subbarao, K.; Shippey, B.M. Hybrid genetic algorithm–collocation method for trajectory optimization. *J. Guid. Control. Dyn.* **2009**, *32*, 1217–1229. [CrossRef]
5. Zhang, D.; Liu, Y. RLV reentry trajectory optimization through hybridization of an improved GA and an SQP algorithm. In Proceedings of the AIAA Guidance, Navigation, and Control Conference, Portland, OR, USA, 8–11 August 2011; AIAA Paper 2011–6658. [CrossRef]
6. NASA. CAPSTONE. Available online: https://www.nasa.gov/mission/capstone/ (accessed on 15 September 2025).
7. NASA. Gateway. Available online: https://www.nasa.gov/mission/gateway/ (accessed on 15 September 2025).
8. Bryson, A.E.; Ho, Y.-C. *Applied Optimal Control: Optimization, Estimation, and Control*; Hemisphere Publishing Corporation/Taylor & Francis: Washington, DC, USA, 1975.
9. Trélat, E.; Zuazua, E. (Eds.) *Numerical Control*; In *Handbook of Numerical Analysis*; Volumes 23 (Part A) and 24 (Part B); Elsevier: Amsterdam, The Netherlands, 2022.
10. NAIF/JPL. *Frames and Coordinate Systems*; Technical report/tutorial; Navigation and Ancillary Information Facility (NAIF), Jet Propulsion Laboratory: Pasadena, CA, USA.
11. Goldberg, D.E. *Genetic Algorithms in Search, Optimization and Machine Learning*; Addison-Wesley: Reading, MA, USA, 1989.
12. Michalewicz, Z. *Genetic Algorithms + Data Structures = Evolution Programs*, 3rd. ed.; Springer: Berlin/Heidelberg, Germany, 1996.
13. García-Martínez, C.; Rodriguez, F.J.; Lozano, M. Genetic Algorithms. In *Handbook of Heuristics*; Martí, R., Pardalos, P. M., Resende, M.G.C., Eds.; Springer International Publishing AG: Cham, Switzerland, 2018; Chapter 15, pp. 431–464.
14. Whitley, D. Next Generation Genetic Algorithms: A User's Guide and Tutorial. In *Handbook of Metaheuristics*; Gendreau, M., Potvin, J.-Y., Eds.; International Series in Operations Research & Management Science; Springer International Publishing AG: Cham, Switzerland, 2019; Volume 272, Chapter 8, pp. 245–274.
15. Nocedal, J.; Wright, S.J. *Numerical Optimization*, 2nd ed.; Springer: New York, NY, USA, 2006; Chapters 12–18 cover SQP and related NLP topics.
16. Belbruno, E.A.; Carrico, J.P. Calculation of weak stability boundary ballistic lunar transfer trajectories. In Proceedings of the AIAA/AAS Astrodynamics Specialist Conference, Denver, CO, USA, 14–17 August 2000.
17. Eiben, A.E.; Smith, J.E. *Introduction to Evolutionary Computing*, 2nd ed.; Natural Computing Series; Springer: Berlin/Heidelberg, Germany, 2015.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.