aerospace

MDPI

Article

Hybrid Optimization Strategy for Time-Optimal Solar Sail Interplanetary Trajectories

Guanwei He [1], Yuan Tan [1], Hao Yuan [1,*], Jie Wang [1] and Zhaokui Wang [2,3,*]

[1] College of Aerospace Science and Engineering, National University of Defense Technology, Changsha 410073, China; gw.he@nudt.edu.cn (G.H.)
[2] School of Aerospace Engineering, Tsinghua University, Beijing 100084, China
[3] State Key Laboratory of Advanced Space Propulsion, Beijing 100084, China
* Correspondence: yuanhao11@nudt.edu.cn (H.Y.); wangzk@tsinghua.edu.cn (Z.W.)

Abstract

Designing time-optimal trajectories for solar sail spacecraft is highly difficult due to the strong nonlinearity of the solar radiation pressure model, the attitude–orbit coupling, and the prevalence of local minima, which often lead to the failure of gradient-based solvers lacking adequate initial guesses. To address this issue, the present study proposes a two-stage hybrid optimization framework: a coarse-search stage uses B-spline-parameterized metaheuristics to identify a dynamically feasible trajectory, which then serves as a physics-informed warm start for direct-collocation-based gradient refinement, strictly satisfying the full nonlinear dynamics and terminal constraints. The methodology is validated through three increasingly difficult space missions: a rendezvous between Earth and Mars (Case A), a near-Earth asteroid rendezvous considering non-ideal optical reflection (Case B), and a Solar Polar Orbiter mission necessitating an 82.75° inclination adjustment with a thermal safety constraint (Case C). Statistical evaluations reveal that no single metaheuristic dominates universally; each algorithm's suitability is contingent on the problem's constraint structure. The hybrid framework further shows a consistent advantage over stand-alone direct collocation: by redirecting the gradient solver toward favorable convergence basins, it locates transfer solutions that remain inaccessible from a cold start.

Keywords: solar sail; trajectory optimization; hybrid optimization; fireworks algorithm; direct collocation

1. Introduction

Solar sailing has evolved from a theoretical curiosity into a proven propulsion technology for deep space exploration [1]. Unlike conventional chemical or electric propulsion systems that rely on expendable propellant, solar sails harness the momentum of solar photons to generate continuous, albeit low, thrust. This propellant-free characteristic theoretically allows for infinite specific impulse, enabling high-energy missions that would be prohibitively expensive or physically impossible with traditional propulsion systems [2,3]. Recent successful technology demonstration missions, such as JAXA's IKAROS [4–6] and The Planetary Society's LightSail 2 [7,8], have validated the fundamental mechanics of solar radiation pressure (SRP) for orbital control, paving the way for future mission concepts ranging from cargo delivery to Mars [9–11] to sample return missions from Near-Earth Asteroids (NEAs) [12–16].

Despite theoretical promise, designing time-optimal trajectories for solar sails remains a formidable challenge in astrodynamics. The difficulty arises from the highly nonlinear

Check for updates

Academic Editor: Vladimir S. Aslanov

Received: 7 July 2026
Revised: 4 August 2026
Accepted: 5 August 2026
Published: 7 August 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.

nature of the SRP force model, which decays with the inverse square of the distance from the Sun, and the complex coupling between the attitude of the spacecraft (cone and clock angles) and its orbital evolution [17]. Numerical optimization methods for such continuous-thrust problems generally fall into two categories: indirect and direct methods. Indirect methods, based on Pontryagin's Minimum Principle, offer high precision but require solving a Two-Point Boundary Value Problem (TPBVP). The convergence domain for these methods is extremely narrow, making the initialization process an art form rather than a systematic procedure [18].

Consequently, direct methods, such as pseudo-spectral methods [19,20] or direct collocation [21,22], have become a standard approach in trajectory design. These methods convert the optimal control problem into a Nonlinear Programming (NLP) problem, which can be solved using gradient-based techniques such as interior-point methods. However, gradient-based solvers are inherently local search algorithms and are notoriously sensitive to the initial guess provided by the user. When applied to complex solar sail trajectories characterized by multiple local minima, such as multi-revolution spirals or the "cranking" maneuvers required for high-inclination solar polar observers [23,24], a "cold start" (using a simple linear interpolation) frequently leads to divergence or convergence to a poor-quality local optimum.

To address similar initialization issues in spacecraft attitude control, Melton [25] proposed a hybrid strategy that combines heuristic algorithms with pseudo-spectral methods. While Melton's work focused on rotational dynamics, the underlying philosophy suggests a promising pathway for solving the more complex translational orbit optimization problem. Considering the inherent initialization challenge in direct collocation methods, this paper proposes a robust, two-stage hybrid optimization framework that integrates the Fireworks Algorithm (FWA) [26] and other heuristic optimization methods [27–29] with a gradient-based solver. The universality and robustness of this framework are demonstrated through three progressively challenging case studies: a standard Earth–Mars rendezvous, a sensitive rendezvous mission to asteroid 2016 HO_3 [30], and a high-inclination Solar Polar Orbiter mission [31], proving the method's capability to automate the solution of long-duration, multi-revolution trajectories that typically defeat traditional cold-start approaches.

The main contributions of this work are as follows: (1) a modular two-stage hybrid framework (Figure 1)—B-spline metaheuristic coarse search coupled, via a physics-informed warm start, to Hermite–Simpson direct-collocation NLP refinement—applicable across single-modal, multi-modal, and path-constrained transfers; (2) the systematic coupling of this scheme with high-fidelity non-ideal solar-sail dynamics for full 3-D interplanetary translational optimization; (3) a three-case benchmark progression supported by independent runs and a statistical apparatus; and (4) a budget-normalization fairness study that isolates the search mechanism from the evaluation budget, yielding problem-class-specific algorithm-selection guidelines.

These contributions distinguish the present work from prior hybrid schemes. Whereas Melton [25] hybridized heuristic search with pseudospectral methods for spacecraft rotational slew, this framework addresses full three-dimensional interplanetary translational optimization under high-fidelity, non-ideal solar-sail dynamics. The novelty here lies not in FWA [26] itself but in the coupling of B-spline metaheuristic global search with direct-collocation refinement and a physics-informed warm start. Finally, in contrast to recent learning-based adaptive control for orbit-attitude takeover based on improved Proximal Policy Optimization [32], this framework remains model-based and gradient-refined, yielding interpretable, certifiable trajectories.

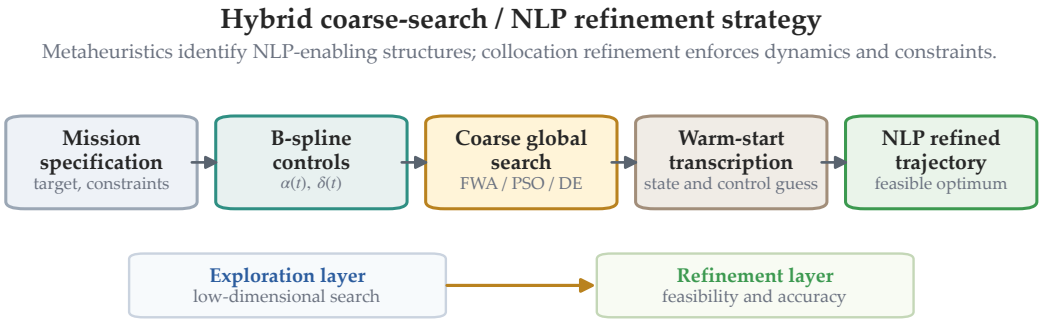


Figure 1. Hybrid coarse-search and NLP-refinement workflow used in this study.

The remainder of this paper is organized as follows: Section 2 details the coordinate frames and solar sail dynamics. Section 3 describes the global search phase using heuristic optimization combined with B-splines structured control profiles. Section 4 outlines the transcription strategy and the direct collocation optimization phase. Section 5 validates the framework through three progressively challenging case studies, Section 6 analyzes the cross-algorithm performance, and Section 7 concludes the paper.

2. Dynamics Modeling

2.1. Coordinate Systems

To describe the spacecraft's trajectory and the attitude–orbit coupling relation, two distinct coordinate systems are introduced: the Heliocentric Inertial frame for numerical integration and a Sun-line reference frame for sail attitude parametrization.

2.1.1. Heliocentric Inertial Frame

The equations of motion are integrated in the Heliocentric Inertial frame ($\mathcal{F}_I$), centered at the Sun (Figure 2). The orthonormal basis $\{\hat{\boldsymbol{I}}, \hat{\boldsymbol{J}}, \hat{\boldsymbol{K}}\}$ is defined such that $\hat{\boldsymbol{I}}$ points toward the Vernal Equinox (γ) at the J2000 epoch, $\hat{\boldsymbol{K}}$ is normal to the mean ecliptic plane of the Earth at J2000, and $\hat{\boldsymbol{J}} = \hat{\boldsymbol{K}} \times \hat{\boldsymbol{I}}$ completes the right-handed system. The spacecraft state vector is $\boldsymbol{X} = [\boldsymbol{r}^\top, \boldsymbol{v}^\top]^\top \in \mathbb{R}^6$, where $\boldsymbol{r} = [x, y, z]^\top$ and $\boldsymbol{v} = [\dot{x}, \dot{y}, \dot{z}]^\top$ represent the heliocentric position and velocity vectors, respectively.

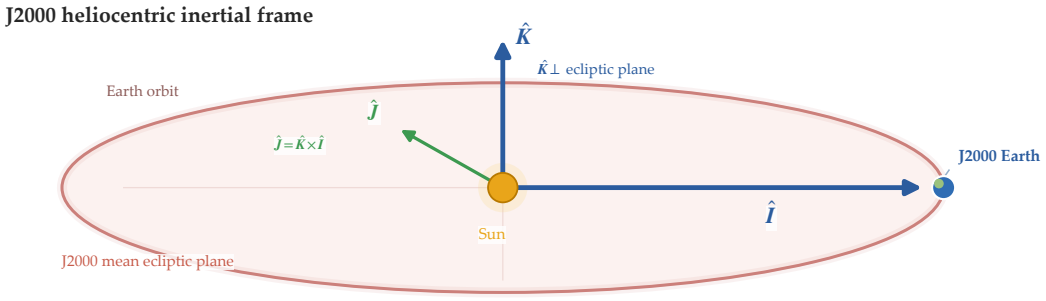


Figure 2. J2000 Heliocentric Inertial reference frame. The $\hat{\boldsymbol{I}}$ axis points toward the J2000 vernal equinox, while $\hat{\boldsymbol{K}}$ is normal to the J2000 mean ecliptic plane of Earth's orbit.

2.1.2. Sun-Line Reference Frame

For the definition of the solar sail's orientation and the resulting thrust direction, a local rotating frame $\mathcal{F}_S(\hat{\boldsymbol{r}}_1, \hat{\boldsymbol{\theta}}_1, \hat{\boldsymbol{\phi}}_1)$ is attached to the spacecraft, following the convention established by Ceriotti [33]. The primary axis $\hat{\boldsymbol{r}}_1 = \boldsymbol{r}_1 / \|\boldsymbol{r}_1\|$ is the unit vector directed along the Sun-sail line, pointing away from the Sun, where $\boldsymbol{r}_1$ denotes the position vector from the Sun to the spacecraft. The second axis $\hat{\boldsymbol{\phi}}_1$ is constructed to be perpendicular to $\hat{\boldsymbol{r}}_1$ and lying in the plane spanned by $\hat{\boldsymbol{r}}_1$ and the orbital angular momentum vector $\boldsymbol{h} = \boldsymbol{r} \times \boldsymbol{v}$, via

orthogonalization, as $\hat{\boldsymbol{\phi}}_1 = (\boldsymbol{h} \times \hat{\boldsymbol{r}}_1)/\|\boldsymbol{h} \times \hat{\boldsymbol{r}}_1\|$. The third axis $\hat{\boldsymbol{\theta}}_1 = \hat{\boldsymbol{\phi}}_1 \times \hat{\boldsymbol{r}}_1$ completes the right-handed triad (Figure 3). This frame rotates with the spacecraft's orbital motion and provides a natural basis for defining the sail orientation relative to the direction of incoming sunlight. The same frame also hosts the sail-surface normal unit vector $\hat{\boldsymbol{n}}$ directed away from the Sun (Figure 3); its parametrization by the pitch and clock angles is developed in Section 2.3.1.

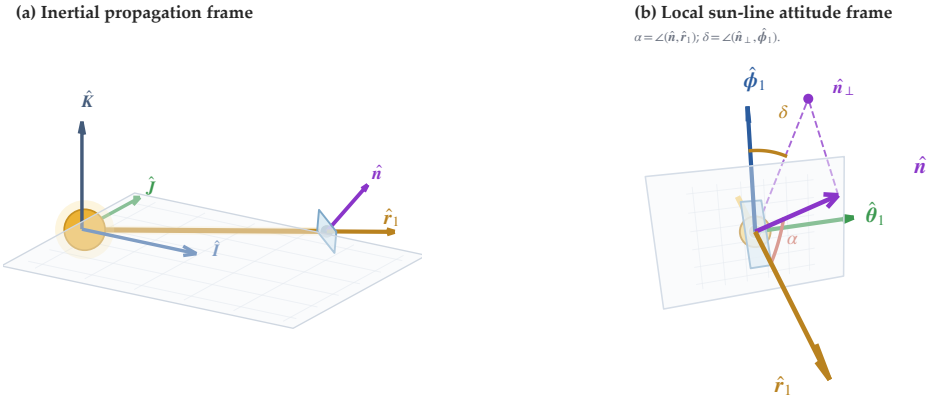


Figure 3. Sun-Line Reference Frame.

2.2. Equations of Motion

The trajectory of the solar sail spacecraft is governed by the interplanetary two-body model perturbed by the Solar Radiation Pressure (SRP). Following the formulation of McInnes [34], the equations of motion in the inertial frame $\mathcal{F}_I$ are

$$\ddot{\boldsymbol{r}} = -\frac{\mu_\odot}{r^3}\boldsymbol{r} + \boldsymbol{a}_s \tag{1}$$

where $\mu_\odot$ is the solar gravitational parameter, $r = \|\boldsymbol{r}\|$ is the heliocentric distance, and $\boldsymbol{a}_s$ is the solar sail acceleration vector. Throughout this study, all dynamical quantities are expressed in Astronomical Units (AUs) for length and solar days (d) for time, yielding $\mu_\odot = 2.9598 \times 10^{-4}$ AU^3 d^{-2}. This unit system is adopted for its natural scaling in heliocentric trajectory problems: distances are of order unity, and time spans are measured in hundreds of days, which improves the numerical conditioning of both the integration and the subsequent optimization.

In first-order state-space form, the system becomes

$$\dot{\boldsymbol{X}} = \begin{bmatrix} \dot{\boldsymbol{r}} \\ \dot{\boldsymbol{v}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{v} \\ -\frac{\mu_\odot}{r^3}\boldsymbol{r} + \boldsymbol{a}_s \end{bmatrix} \tag{2}$$

This Cartesian formulation avoids the singularities associated with classical Keplerian elements for circular or equatorial orbits and is well-suited for numerical integration of non-Keplerian trajectories. The state equations are propagated using an explicit Runge–Kutta method of order 4(5) (RK45) with adaptive step-size control, where the local truncation error tolerances are set to 10^{-8} (absolute) and 10^{-10} (relative) to ensure high-fidelity propagation over the multi-year flight durations considered in this work. The sail acceleration $\boldsymbol{a}_s$ depends on both the sail's optical properties and its instantaneous attitude.

2.3. Solar Radiation Pressure Force Model

To capture the realistic performance of the solar sail, this study employs the non-ideal optical reflectance model developed by McInnes [34]. While the ideal model assumes the sail to be a perfect specular reflector, this optical model also accounts for the effects of

absorption, diffuse reflection, and thermal emission from both surfaces of the sail film. Both models can be captured within a unified mathematical framework, as described below.

2.3.1. Sail Attitude Definition

The sail normal vector $\hat{\boldsymbol{n}}$ is parametrized by two control angles defined with respect to the Sun-line frame $\mathcal{F}_S(\hat{\boldsymbol{r}}_1, \hat{\boldsymbol{\theta}}_1, \hat{\boldsymbol{\phi}}_1)$. The pitch angle α is the angle between the sail normal $\hat{\boldsymbol{n}}$ and the Sun-line direction $\hat{\boldsymbol{r}}_1$, while the clock angle δ is the angle between the projection of $\hat{\boldsymbol{n}}$ onto the $(\hat{\boldsymbol{\theta}}_1, \hat{\boldsymbol{\phi}}_1)$-plane and the $\hat{\boldsymbol{\phi}}_1$ axis. With this convention, the sail normal vector expressed in frame $\mathcal{F}_S$ takes the form:

$$\hat{\boldsymbol{n}}\big|_{\mathcal{F}_S} = \begin{bmatrix} \cos\alpha \\ \sin\alpha\sin\delta \\ \sin\alpha\cos\delta \end{bmatrix} \tag{3}$$

Since the solar sail cannot generate an acceleration component directed toward the Sun (i.e., the normal must always maintain a positive projection along the Sun-line), the admissible ranges of the control angles are $0° \leqslant \alpha \leqslant 90°$ and $-180° \leqslant \delta \leqslant 180°$.

2.3.2. Normal and Tangential Acceleration Components

The solar-sail acceleration vector $\boldsymbol{a}_s$ can be decomposed into a component normal to the sail surface, $a_n\hat{\boldsymbol{n}}$, and a component tangential to the sail, $a_t\hat{\boldsymbol{t}}$:

$$\boldsymbol{a}_s = a_n\hat{\boldsymbol{n}} + a_t\hat{\boldsymbol{t}} = a_s\hat{\boldsymbol{m}} \tag{4}$$

where $\hat{\boldsymbol{n}}$ is the unit vector normal to the sail surface pointing away from the Sun, and $\hat{\boldsymbol{t}}$ is the unit vector tangential to the sail surface lying in the plane formed by the Sun-line $\hat{\boldsymbol{r}}_1$ and $\hat{\boldsymbol{n}}$. The resultant acceleration has magnitude $a_s = \sqrt{a_n^2 + a_t^2}$ and direction $\hat{\boldsymbol{m}}$.

The magnitudes of the normal and tangential acceleration components are derived from the optical properties of the sail material. Following the formulation by Heiligers [30], the expressions are

$$\begin{aligned} a_n = \frac{1}{2}\beta\frac{1-\mu_\odot}{r_1^2}\Big[&(1+\tilde{r}s)\cos^2\alpha + B_f(1-s)\tilde{r}\cos\alpha \\ &+ (1-\tilde{r})\frac{\epsilon_f B_f - \epsilon_b B_b}{\epsilon_f + \epsilon_b}\cos\alpha\Big] \end{aligned} \tag{5}$$

$$a_t = \frac{1}{2}\beta\frac{1-\mu_\odot}{r_1^2}(1-\tilde{r}s)\cos\alpha\sin\alpha \tag{6}$$

In Equations (5) and (6), β is the solar-sail lightness number, a dimensionless parameter representing the ratio of the maximum SRP acceleration to the local solar gravitational acceleration, defined as $\beta = \sigma^*/\sigma$, where σ denotes the sail loading (total spacecraft mass divided by sail area, in $\mathrm{g/m^2}$) and $\sigma^* \approx 1.53\ \mathrm{g/m^2}$ is the critical sail loading parameter. The quantity $r_1 = \|\boldsymbol{r}_1\|$ is the Sun–spacecraft distance.

The optical properties appearing in the above expressions are as follows. The reflectivity coefficient $\tilde{r}$ indicates the fraction of incident photons that are reflected, with the remaining fraction $(1-\tilde{r})$ being absorbed. Of the reflected photons, a fraction s undergoes specular reflection while the complementary fraction $(1-s)$ is reflected diffusely in a Lambertian distribution. The non-Lambertian coefficients B_f and B_b characterize the angular distribution of the diffusely reflected and thermally emitted photons from the front and back surfaces of the sail, respectively, accounting for deviations from a perfectly Lambertian

pattern. Finally, ϵ_f and ϵ_b are the emissivity coefficients of the front and back surfaces, governing the thermal re-emission of absorbed solar energy from each face.

The physical structure of the normal acceleration a_n merits careful examination. The first term, $(1 + \tilde{r}s)\cos^2\alpha$, represents the combined momentum transfer from the incident photon flux and the specularly reflected photons; for an ideal sail with $\tilde{r} = 1$ and $s = 1$, this reduces to $2\cos^2\alpha$, recovering the classical ideal model. The second term, $B_f(1 - s)\tilde{r}\cos\alpha$, captures the normal force component arising from diffusely reflected photons on the front surface and vanishes identically for a perfectly specular sail ($s = 1$). The third term, $(1 - \tilde{r})(\epsilon_f B_f - \epsilon_b B_b)/(\epsilon_f + \epsilon_b)\cos\alpha$, represents the net normal force due to the differential thermal emission between the front and back surfaces; if the emission characteristics are symmetric ($\epsilon_f B_f = \epsilon_b B_b$), this contribution vanishes.

The tangential acceleration a_t arises solely from non-specular effects. Its pre-factor $(1 - \tilde{r}s)$ ensures that for the ideal sail model ($\tilde{r} = 1$, $s = 1$), $a_t = 0$ identically, confirming that the ideal model produces only a force normal to the sail surface. The numerical values of these optical coefficients for the two reflectance models considered in this study are summarized in Table 1.

Table 1. Optical coefficients for ideal and non-ideal solar sail reflectance models [35].

Reflectance Model	$\tilde{r}$	s	B_f	B_b	ϵ_f	ϵ_b
Ideal	1	1	—	—	—	—
Optical (NEA Scout)	0.91	0.94	0.79	0.67	0.025	0.27

2.3.3. Centerline Lag and Force Direction

A critical consequence of the non-ideal model is that the resultant acceleration unit vector $\hat{\boldsymbol{m}}$ is not aligned with the sail normal $\hat{\boldsymbol{n}}$. The geometry of the sail-normal direction, tangential direction, and resultant acceleration is illustrated in Figure 4. The tangential force component a_t arising from imperfect reflection rotates the acceleration vector away from the normal by an auxiliary angle φ, commonly referred to as the "centerline lag":

$$\varphi = \arctan\left(\frac{a_t}{a_n}\right) \tag{7}$$

The effective cone angle θ of the resultant thrust vector, measured from the Sun-line $\hat{\boldsymbol{r}}_1$, is therefore related to the sail attitude by

$$\theta = \alpha - \varphi \tag{8}$$

This distinction between the attitude angle α, which the spacecraft attitude control system commands, and the force angle θ, which governs the orbital dynamics, is a defining feature of the non-ideal model and is essential for high-fidelity trajectory optimization. The direction of the resultant acceleration $\hat{\boldsymbol{m}}$ in the Sun-line frame $\mathcal{F}_S$ is then expressed as

$$\hat{\boldsymbol{m}}\big|_{\mathcal{F}_S} = \begin{bmatrix} \cos\theta \\ \sin\theta\sin\delta \\ \sin\theta\cos\delta \end{bmatrix} \tag{9}$$

For the ideal model, $\varphi = 0$ and $\theta = \alpha$, so the force direction $\hat{\boldsymbol{m}}$ coincides exactly with the sail normal $\hat{\boldsymbol{n}}$, as expected.

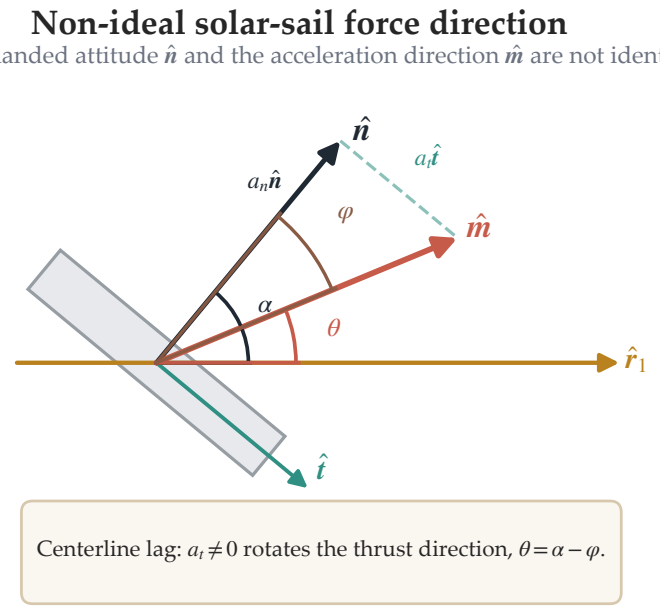


Figure 4. Non-ideal solar-sail force direction and the centerline lag between the commanded sail normal and the resultant acceleration direction.

2.3.4. Special Case: Ideal Reflectance Model

For validation and benchmarking purposes, this study also considers the ideal reflectance model as a limiting case. Substituting $\tilde{r} = 1$ and $s = 1$ into the general expressions of Equations (5) and (6) yields

$$a_n^{\text{ideal}} = \beta \frac{1 - \mu_\odot}{r_1^2} \cos^2 \alpha, \quad a_t^{\text{ideal}} = 0 \tag{10}$$

and therefore $\boldsymbol{a}_s = a_n^{\text{ideal}} \hat{\boldsymbol{n}}$, which is the well-known ideal solar radiation model producing a force purely along the sail normal direction. In this case, the sail attitude directly determines the thrust direction without any centerline lag: $\hat{\boldsymbol{m}} = \hat{\boldsymbol{n}}$ and $\theta = \alpha$.

2.4. Coordinate Transformation

The optimization variables are the control angles $\boldsymbol{u} = [\alpha, \delta]^\top$, defined in the Sun-line frame $\mathcal{F}_S$. To integrate the equations of motion in $\mathcal{F}_I$, the acceleration vector must be transformed from $\mathcal{F}_S$ to $\mathcal{F}_I$. The rotation matrix $\boldsymbol{R}_{\mathcal{S}\to\mathcal{I}}$ is constructed by assembling the basis vectors of $\mathcal{F}_S$, expressed in inertial coordinates, as column vectors:

$$\boldsymbol{R}_{\mathcal{S}\to\mathcal{I}} = \begin{bmatrix} \hat{\boldsymbol{r}}_1 & \hat{\boldsymbol{\theta}}_1 & \hat{\boldsymbol{\phi}}_1 \end{bmatrix} \tag{11}$$

The complete solar sail acceleration vector in the inertial frame, to be substituted into the equations of motion, is obtained by combining the scalar force magnitudes from Equations (5) and (6), the direction vector from Equation (9), and the rotation matrix formula, Equation (11):

$$\boldsymbol{a}_s = \underbrace{\sqrt{a_n^2 + a_t^2}}_{\text{magnitude}} \boldsymbol{R}_{\mathcal{S}\to\mathcal{I}} \, \hat{\boldsymbol{m}}\big|_{\mathcal{F}_S} \tag{12}$$

Expanding this expression explicitly:

$$\boldsymbol{a}_s = \sqrt{a_n^2 + a_t^2} \begin{bmatrix} \hat{\boldsymbol{r}}_1 & \hat{\boldsymbol{\theta}}_1 & \hat{\boldsymbol{\phi}}_1 \end{bmatrix} \begin{bmatrix} \cos\theta \\ \sin\theta \sin\delta \\ \sin\theta \cos\delta \end{bmatrix} \tag{13}$$

where $\theta = \alpha - \arctan(a_t/a_n)$, and the normal and tangential components a_n and a_t are functions of the pitch angle α and the optical coefficients as defined.

This formulation completes the dynamical modeling chain: given the instantaneous state $(\boldsymbol{r}, \boldsymbol{v})$ and the control input (α, δ), the inertial acceleration $\boldsymbol{a}_s$ can be evaluated deterministically, enabling forward integration of the trajectory with full accounting for the optical imperfections of the solar sail material.

3. Global Search Methodology

Optimizing interplanetary trajectories for solar sail spacecraft is inherently a continuous, infinite-dimensional optimal control problem. To make this computationally tractable, the control profile is discretized via direct transcription. Specifically, cubic B-splines are used to parameterize the sail attitude angles, converting the system into a finite-dimensional Nonlinear Programming (NLP) formulation. The parameter space is then searched using three population-based metaheuristic algorithms: the Fireworks Algorithm (FWA), Particle Swarm Optimization (PSO), and Differential Evolution (DE). These methods are chosen for their effectiveness in avoiding local minima in multi-modal solution space—a crucial requirement when handling the complex path constraints and control structures in trajectory design.

3.1. B-Spline Parameterization for Control Variables

To preserve the physical continuity of the control variables while minimizing the dimensionality of the search space, the time histories of the pitch angle $\alpha(t)$ and clock angle $\delta(t)$ are approximated by B-spline basis functions.

A B-spline of degree p is defined recursively over a knot vector $\boldsymbol{T} = \{t_0, t_1, \ldots, t_{n+p+1}\}$. For a uniformly spaced knot sequence spanning the flight time $t \in [t_0, t_f]$, the i-th B-spline basis function $N_{i,p}(t)$ is computed using the Cox–de Boor recurrence relation. The zeroth-degree basis functions are piecewise constants:

$$N_{i,0}(t) = \begin{cases} 1 & \text{if } t_i \leq t < t_{i+1} \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

For degrees $p \geq 1$, the basis functions satisfy:

$$N_{i,p}(t) = \frac{t - t_i}{t_{i+p} - t_i} N_{i,p-1}(t) + \frac{t_{i+p+1} - t}{t_{i+p+1} - t_{i+1}} N_{i+1,p-1}(t) \tag{15}$$

In this work, cubic B-splines ($p = 3$) are employed, ensuring C^2 continuity of the control profile. This smoothness property is physically motivated: it prevents unrealistic discontinuities in the commanded sail orientation that would violate actuator slew-rate limits in a practical implementation. The control functions $\alpha(t)$ and $\delta(t)$ are expressed as linear combinations of the B-spline basis functions:

$$\alpha(t) = \sum_{i=1}^{n_\alpha} P_{\alpha,i} N_{i,3}(t), \quad \delta(t) = \sum_{j=1}^{n_\delta} P_{\delta,j} N_{j,3}(t) \tag{16}$$

where $P_{\alpha,i}$ and $P_{\delta,j}$ are the control angle value to be determined by the optimizer, and n_α and n_δ denote the number of control points for each angle. The optimization design vector $\boldsymbol{D}$ aggregates these coefficients along with the time of flight (if variable):

$$\boldsymbol{D} = [P_{\alpha,1}, \ldots, P_{\alpha,n_\alpha}, P_{\delta,1}, \ldots, P_{\delta,n_\delta}, t_f]^\top \in \mathbb{R}^{2n+1} \tag{17}$$

where $n = n_\alpha = n_\delta$ for simplicity. The bounds on $\boldsymbol{D}$ are inherited directly from the physical constraints established:

$$\boldsymbol{D}_{\min} \leq \boldsymbol{D} \leq \boldsymbol{D}_{\max} \tag{18}$$

with $P_{\alpha,i} \in [0°, 90°]$ and $P_{\delta,j} \in [-180°, 180°]$ for all i, j. Given a candidate design vector $\boldsymbol{D}$, the control history is reconstructed via Equation (16), and the state differential equations Equation (2) are integrated forward in time using the RK45 scheme.

3.2. Fireworks Algorithm

The Fireworks Algorithm (FWA) [36] allocates, per firework $\boldsymbol{X}_i$, a spark count S_i and explosion amplitude A_i inversely proportional to fitness:

$$S_i = \left\lfloor M \cdot \frac{Y_{\max} - J(\boldsymbol{X}_i) + \epsilon}{\sum_{k=1}^{N_{\text{fw}}} (Y_{\max} - J(\boldsymbol{X}_k)) + \epsilon} \right\rfloor \tag{19}$$

$$A_i = \hat{A} \cdot \frac{J(\boldsymbol{X}_i) - Y_{\min} + \epsilon}{\sum_{k=1}^{N_{\text{fw}}} (J(\boldsymbol{X}_k) - Y_{\min}) + \epsilon} \tag{20}$$

Sparks are generated by perturbing randomly selected dimensions:

$$x_{\text{spark}}^k = x_i^k + A_i \cdot \mathcal{U}(-1, 1), \tag{21}$$

supplemented by a small number of Gaussian mutation sparks. Out-of-bounds sparks follow the unified boundary-handling rule of Section 3.5; elitist selection among the combined firework–spark pool ensures monotonic improvement. Algorithm 1 summarizes the full procedure.

Algorithm 1: Fireworks Algorithm (FWA)

Input: Population size N_{fw}, spark budget M, max amplitude $\hat{A}$, max iterations N_{iter}
Output: Best solution $\boldsymbol{X}_{\text{best}}$
Initialize population $\{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_{N_{\text{fw}}}\}$ uniformly in $[\boldsymbol{D}_{\min}, \boldsymbol{D}_{\max}]$;
Evaluate $J(\boldsymbol{X}_i)$ for $i = 1, \ldots, N_{\text{fw}}$;
$\boldsymbol{X}_{\text{best}} \leftarrow \arg\min_i J(\boldsymbol{X}_i)$;
for $iter = 1$ **to** N_{iter} **do**
  Compute $Y_{\max} = \max_i J(\boldsymbol{X}_i)$, $Y_{\min} = \min_i J(\boldsymbol{X}_i)$;
  for $i = 1$ **to** N_{fw} **do**
    Compute S_i using Equation (19); compute A_i using Equation (20);
    for $s = 1$ **to** S_i **do**
      Generate explosion spark $\boldsymbol{X}_{\text{spark}}$ using Equation (21); apply boundary handling (Section 3.5); evaluate $J(\boldsymbol{X}_{\text{spark}})$;
    Generate $\tilde{S}$ Gaussian mutation sparks;
  Pool $\leftarrow$ all fireworks and sparks;
  Sort Pool by J in ascending order;
  Select top N_{fw} solutions as new fireworks $\{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_{N_{\text{fw}}}\}$;
  Update $\boldsymbol{X}_{\text{best}}$ if improved;
return $\boldsymbol{X}_{best}$

3.3. Particle Swarm Optimization

Particle Swarm Optimization (PSO) [37] models social foraging, in which each particle's velocity update combines inertial, cognitive (c_1), and social (c_2) terms:

$$\boldsymbol{V}_i^{k+1} = w\boldsymbol{V}_i^k + c_1 r_1 \odot (\boldsymbol{P}_{\text{best},i} - \boldsymbol{X}_i^k) + c_2 r_2 \odot (\boldsymbol{G}_{\text{best}} - \boldsymbol{X}_i^k), \tag{22}$$

where $\boldsymbol{P}_{\text{best},i}$ and $\boldsymbol{G}_{\text{best}}$ denote the personal and global best positions. The inertia weight w decays linearly from 0.9 to 0.4 via

$$w(k) = w_{\text{init}} - \frac{(w_{\text{init}} - w_{\text{final}})(k-1)}{N_{\text{iter}} - 1}, \tag{23}$$

and positions advance via

$$\boldsymbol{X}_i^{k+1} = \boldsymbol{X}_i^k + \boldsymbol{V}_i^{k+1}. \tag{24}$$

Out-of-bounds positions follow Section 3.5. Personal and global bests are updated greedily. Algorithm 2 summarizes the procedure.

Algorithm 2: Particle Swarm Optimization (PSO)

Input: Population size N_p, acceleration coefficients c_1, c_2, inertia $w_{\text{init}}, w_{\text{final}}$, max iterations N_{iter}
Output: Best solution $\boldsymbol{G}_{\text{best}}$
Initialize positions $\{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_{N_p}\}$ uniformly in $[\boldsymbol{D}_{\min}, \boldsymbol{D}_{\max}]$;
Initialize velocities $\{\boldsymbol{V}_1, \ldots, \boldsymbol{V}_{N_p}\}$ uniformly in $[-0.5(\boldsymbol{D}_{\max} - \boldsymbol{D}_{\min}),\ 0.5(\boldsymbol{D}_{\max} - \boldsymbol{D}_{\min})]$;
Evaluate $J(\boldsymbol{X}_i)$ for $i = 1, \ldots, N_p$;
Set $\boldsymbol{P}_{\text{best},i} \leftarrow \boldsymbol{X}_i$ for all i;
$\boldsymbol{G}_{\text{best}} \leftarrow \arg\min_i J(\boldsymbol{P}_{\text{best},i})$;
for $k = 1$ **to** N_{iter} **do**
  Compute $w(k)$ using Equation (23);
  for $i = 1$ **to** N_p **do**
    Generate $r_1, r_2 \sim \mathcal{U}(0,1)^{\dim(\boldsymbol{D})}$;
    Update velocity using Equation (22); update position using Equation (24); apply boundary handling (Section 3.5) to $\boldsymbol{X}_i^{k+1}$;
    Evaluate $J(\boldsymbol{X}_i^{k+1})$;
    if $J(\boldsymbol{X}_i^{k+1}) < J(\boldsymbol{P}_{best,i})$ **then**
      $\boldsymbol{P}_{\text{best},i} \leftarrow \boldsymbol{X}_i^{k+1}$;
    if $J(\boldsymbol{X}_i^{k+1}) < J(\boldsymbol{G}_{best})$ **then**
      $\boldsymbol{G}_{\text{best}} \leftarrow \boldsymbol{X}_i^{k+1}$;
return $\boldsymbol{G}_{best}$

3.4. Differential Evolution

Differential Evolution (DE/rand/1/bin) [38] operates through mutation, binomial crossover, and greedy selection. The mutation operator constructs a donor vector by adding a scaled difference of two randomly selected population members to a third:

$$\boldsymbol{V}_{i,G} = \boldsymbol{X}_{r_1,G} + F \cdot (\boldsymbol{X}_{r_2,G} - \boldsymbol{X}_{r_3,G}), \tag{25}$$

with scaling factor $F = 0.8$. Binomial crossover

$$U_{i,G}^j = \begin{cases} V_{i,G}^j & \text{if } \text{rand}_j \le CR \text{ or } j = j_{\text{rand}} \\ X_{i,G}^j & \text{otherwise} \end{cases} \tag{26}$$

mixes the donor and target vectors with $CR = 0.9$ to produce a trial vector $\boldsymbol{U}_{i,G}$. Greedy selection

$$\boldsymbol{X}_{i,G+1} = \begin{cases} \boldsymbol{U}_{i,G} & \text{if } J(\boldsymbol{U}_{i,G}) \leq J(\boldsymbol{X}_{i,G}) \\ \boldsymbol{X}_{i,G} & \text{otherwise} \end{cases} \tag{27}$$

retains the better of $\boldsymbol{U}_{i,G}$ and $\boldsymbol{X}_{i,G}$. Out-of-bounds trial vectors follow Section 3.5. Algorithm 3 details the full procedure.

Algorithm 3: Differential Evolution (DE/rand/1/bin)

Input: Population size N_p, scaling factor F, crossover rate CR, max generations N_{gen}

Output: Best solution $\boldsymbol{X}_{\text{best}}$

Initialize population $\{\boldsymbol{X}_{1,0}, \ldots, \boldsymbol{X}_{N_p,0}\}$ uniformly in $[\boldsymbol{D}_{\min}, \boldsymbol{D}_{\max}]$;
Evaluate $J(\boldsymbol{X}_{i,0})$ for $i = 1, \ldots, N_p$;
$\boldsymbol{X}_{\text{best}} \leftarrow \arg\min_i J(\boldsymbol{X}_{i,0})$;
for $G = 0$ **to** $N_{gen} - 1$ **do**
  for $i = 1$ **to** N_p **do**
    Select distinct indices $r_1, r_2, r_3 \in \{1, \ldots, N_p\} \setminus \{i\}$ randomly;
    Generate donor vector using Equation (25):
    $\boldsymbol{V}_{i,G} \leftarrow \boldsymbol{X}_{r_1,G} + F \cdot (\boldsymbol{X}_{r_2,G} - \boldsymbol{X}_{r_3,G})$;
    Select $j_{\text{rand}} \sim \text{Uniform}\{1, \ldots, \dim(\boldsymbol{D})\}$;
    for $j = 1$ **to** $\dim(\boldsymbol{D})$ **do**
      Generate $\text{rand}_j \sim \mathcal{U}(0, 1)$;
      if $rand_j \leq CR$ *or* $j = j_{rand}$ **then**
        $U^j_{i,G} \leftarrow V^j_{i,G}$;
      else
        $U^j_{i,G} \leftarrow X^j_{i,G}$;
    Apply boundary handling (Section 3.5) to $\boldsymbol{U}_{i,G}$ (if needed);
    Evaluate $J(\boldsymbol{U}_{i,G})$;
    Apply selection using Equation (27):;
    if $J(\boldsymbol{U}_{i,G}) \leq J(\boldsymbol{X}_{i,G})$ **then**
      $\boldsymbol{X}_{i,G+1} \leftarrow \boldsymbol{U}_{i,G}$;
    else
      $\boldsymbol{X}_{i,G+1} \leftarrow \boldsymbol{X}_{i,G}$;
    if $J(\boldsymbol{X}_{i,G+1}) < J(\boldsymbol{X}_{best})$ **then**
      $\boldsymbol{X}_{\text{best}} \leftarrow \boldsymbol{X}_{i,G+1}$;
return $\boldsymbol{X}_{best}$

3.5. Boundary Handling

All three metaheuristics return out-of-bounds candidates to the feasible box through the same modulo-wrapping rule. Whenever a generated candidate (FWA spark, PSO particle, or DE trial vector) exceeds a bound, the offending component is mapped as

$$x^k \leftarrow D^k_{\min} + |x^k - D^k_{\min}| \bmod (D^k_{\max} - D^k_{\min}), \tag{28}$$

where x^k is the k-th component of the candidate. This rule preserves the distance traveled by the displaced candidate and re-injects it on the opposite side of the search domain, sustaining population diversity.

This completes the description of the optimization framework. The combination of B-spline parameterization and population-based metaheuristics provides a robust methodology for the global search of time-optimal solar-sail trajectories, avoiding the pitfalls of gradient-based local optimizers while maintaining computational efficiency through adaptive exploration strategies.

4. Trajectory Refinement via Gradient-Based Optimization

The global search methodology presented in Section 3 provides a quasi-optimal trajectory by parameterizing control profile with cubic B-splines and exploring the finite-dimensional design space via population-based metaheuristics. While this approach effectively identifies the proximity of the optimum, the solution has two limitations: the B-spline smoothness confines control to a low-dimensional manifold, preventing recovery of fine-scale maneuvers (e.g., rapid attitude changes near perihelion), and the metaheuristics' penalty-based constraint handling does not enforce strict terminal constraints. To address these issues, this section presents stage two of the hybrid framework: a gradient-based refinement that transcribes the continuous optimal control problem into a large-scale NLP via direct collocation and solves it with an interior-point method initialized from the first stage.

4.1. Direct Collocation Transcription via the Hermite–Simpson Scheme

In contrast to shooting methods, which propagate the state forward in time and are sensitive to errors that accumulate over long arcs, collocation methods treat both the state and control variables at each node as independent decision variables and impose the dynamics as algebraic equality constraints. This simultaneous approach yields a sparse constraint Jacobian and is particularly well-suited for the long-duration, multi-revolution solar sail transfers considered in this work.

The total flight time $[t_0, t_f]$ is divided into N equally spaced sub-intervals by introducing $N+1$ mesh nodes $\{t_0, t_1, \ldots, t_N\}$, where $t_k = t_0 + k \cdot h$ and $h = (t_f - t_0)/N$ is the uniform step size. At each node t_k, the state vector $\boldsymbol{X}_k = [\boldsymbol{r}_k^\top, \boldsymbol{v}_k^\top]^\top \in \mathbb{R}^6$ and the control vector $\boldsymbol{u}_k = [\alpha_k, \delta_k]^\top \in \mathbb{R}^2$ are treated as independent optimization variables. The complete NLP decision vector is assembled as

$$\boldsymbol{Z} = [\boldsymbol{X}_0^\top, \boldsymbol{u}_0^\top, \boldsymbol{X}_1^\top, \boldsymbol{u}_1^\top, \ldots, \boldsymbol{X}_N^\top, \boldsymbol{u}_N^\top, t_f]^\top \in \mathbb{R}^{8(N+1)+1} \tag{29}$$

The dynamical constraints are enforced using the Hermite–Simpson collocation scheme, which provides fourth-order accuracy. For each sub-interval $[t_k, t_{k+1}]$, a collocation point is introduced at the midpoint $t_{k+1/2} = (t_k + t_{k+1})/2$. The state and its time derivative at the collocation point are approximated using cubic Hermite interpolation:

$$\boldsymbol{X}_{k+1/2} = \frac{1}{2}(\boldsymbol{X}_k + \boldsymbol{X}_{k+1}) + \frac{h}{8}(\boldsymbol{f}_k - \boldsymbol{f}_{k+1}) \tag{30}$$

where $\boldsymbol{f}_k \equiv \boldsymbol{f}(t_k, \boldsymbol{X}_k, \boldsymbol{u}_k)$ denotes the right-hand side of the equations of motion (Equation (2)) evaluated at node k. The control at the midpoint is obtained by linear interpolation:

$$\boldsymbol{u}_{k+1/2} = \frac{1}{2}(\boldsymbol{u}_k + \boldsymbol{u}_{k+1}) \tag{31}$$

The dynamics at the collocation point are then evaluated as

$$\boldsymbol{f}_{k+1/2} = \boldsymbol{f}(t_{k+1/2}, \boldsymbol{X}_{k+1/2}, \boldsymbol{u}_{k+1/2}) \tag{32}$$

The Simpson quadrature rule yields the defect constraint vector for the k-th segment:

$$\boldsymbol{\zeta}_k = \boldsymbol{X}_{k+1} - \boldsymbol{X}_k - \frac{h}{6}(\boldsymbol{f}_k + 4\boldsymbol{f}_{k+1/2} + \boldsymbol{f}_{k+1}) = \mathbf{0} \tag{33}$$

for $k = 0, 1, \ldots, N-1$. This yields a total of $6N$ scalar equality constraints arising from the dynamics. The fourth-order accuracy of the Simpson rule ensures that, for sufficiently fine grids, the discretization error is negligible compared to the convergence tolerances of the NLP solver.

4.2. Warm-Start Strategy: From B-Spline Solution to NLP Initial Guess

The efficacy of the gradient-based refinement is highly sensitive to the quality of the initial guess $\boldsymbol{Z}_0$ supplied to the NLP solver. When applied to solar sail spacecraft trajectory problems—particularly those involving the non-ideal optical force model, which introduces tangential acceleration components that significantly complicate the dynamics—a cold start (e.g., linear interpolation between boundary states) almost invariably leads to divergence. The present subsection details a systematic procedure for constructing a high-quality warm start by mapping the B-spline solution obtained from the global search phase onto the collocation grid used by the NLP formulation.

Let $\boldsymbol{D}^* = [P^*_{\alpha,1}, \ldots, P^*_{\alpha,n}, P^*_{\delta,1}, \ldots, P^*_{\delta,n}, t^*_f]^\top$ denote the optimal design vector returned by the metaheuristic algorithm in Section 3. The continuous control profiles are reconstructed from the B-spline representation:

$$\alpha^*(t) = \sum_{i=1}^{n} P^*_{\alpha,i} N_{i,3}(t), \quad \delta^*(t) = \sum_{j=1}^{n} P^*_{\delta,j} N_{j,3}(t) \tag{34}$$

These profiles are evaluated at each node of the collocation grid to produce the initial control guess:

$$\boldsymbol{u}_k^{(0)} = [\alpha^*(t_k),\ \delta^*(t_k)]^\top, \quad k = 0, 1, \ldots, N \tag{35}$$

Given the sampled control history $\{\boldsymbol{u}_k^{(0)}\}_{k=0}^{N}$ and the known departure state $\boldsymbol{X}_0 = \boldsymbol{X}_{\text{Earth}}(t_0)$, the initial state guess is reconstructed by forward numerical integration of the equations of motion (Equation (2)) using the same RK45 scheme employed in the global search stage. The integrated state is recorded at each node:

$$\boldsymbol{X}_k^{(0)} = \boldsymbol{\Phi}(t_0, t_k; \{\boldsymbol{u}^{(0)}\}), \quad k = 0, 1, \ldots, N \tag{36}$$

where $\boldsymbol{\Phi}$ denotes the numerical flow map. Together with the initial flight time $t_f^{(0)} = t_f^*$, the complete initial guess for the NLP solver is assembled as

$$\boldsymbol{Z}_0 = [(\boldsymbol{X}_0^{(0)})^\top, (\boldsymbol{u}_0^{(0)})^\top, \ldots, (\boldsymbol{X}_N^{(0)})^\top, (\boldsymbol{u}_N^{(0)})^\top, t_f^{(0)}]^\top \tag{37}$$

By construction, this initial guess ensures that the state trajectory satisfies the equations of motion to within the integration tolerance. Consequently, the initial defect vectors $\boldsymbol{\zeta}_k$ are near-zero, placing $\boldsymbol{Z}_0$ in the immediate neighborhood of the feasible manifold. Because the solver starts close to a dynamically consistent trajectory, it is substantially more likely to iterate within its convergence basin; this is corroborated by the cold-start comparison of Section 6.4, where warm starts reached a superior basin that cold starts could not.

4.3. NLP Problem Formulation

With the transcription and warm-start strategies established, the continuous optimal control problem is now converted into a structured NLP. The formulation is presented in standard form to facilitate implementation in existing NLP solver frameworks.

The primary performance metric adopted in this study is the minimization of the total flight time:

$$\min_{\boldsymbol{Z}} \; J(\boldsymbol{Z}) = t_f \tag{38}$$

This choice is motivated by the practical importance of mission duration in solar sail trajectory design, where the continuous low-thrust nature of the propulsion demands flight times spanning hundreds to thousands of days.

The equality constraints consist of the $6N$ defect equations from Equation (33), enforcing dynamical consistency across all N segments. The departure state is fixed at the Earth's ephemeris position and velocity at the launch epoch t_0:

$$\boldsymbol{X}_0 = \boldsymbol{X}_{\text{Earth}}(t_0) \tag{39}$$

For rendezvous missions, both position and velocity must match those of the target body at arrival:

$$\boldsymbol{r}_N = \boldsymbol{r}_{\text{target}}(t_f), \quad \boldsymbol{v}_N = \boldsymbol{v}_{\text{target}}(t_f) \tag{40}$$

yielding six additional equality constraints. The total number of equality constraints is therefore $6N + 12$.

Regarding inequality constraints, the physical limitations of the solar sail impose box constraints on the control variables at each node:

$$0^\circ \leqslant \alpha_k \leqslant 90^\circ, \quad -180^\circ \leqslant \delta_k \leqslant 180^\circ, \quad k = 0, \ldots, N \tag{41}$$

For missions involving close solar approaches (e.g., the Solar Polar Orbiter mission), a minimum heliocentric distance constraint is imposed to respect the thermal limitations of the sail material:

$$\|\boldsymbol{r}_k\| \geqslant r_{\min}, \quad k = 0, \ldots, N \tag{42}$$

where $r_{\min}$ is a mission-dependent parameter (typically 0.2–0.5 AU). Additionally, the flight time is bounded within a physically meaningful range:

$$t_{f,\min} \leqslant t_f \leqslant t_{f,\max} \tag{43}$$

4.4. Interior-Point Method Solver Configuration

The NLP problem defined above is solved using an interior-point method (IPM), as implemented in MATLAB's `fmincon` solver with the `interior-point` algorithm option. The IPM approaches the constrained optimum by solving a sequence of barrier subproblems in which the inequality constraints are incorporated into the objective function via logarithmic barrier terms. At each iteration, the algorithm solves the Karush–Kuhn–Tucker (KKT) system to compute a Newton-like search direction, and a line search with a merit function ensures sufficient decrease in both the objective and constraint violation.

The convergence tolerances are set as follows: the optimality tolerance (first-order KKT residual) is 10^{-8}, the constraint violation tolerance is 10^{-10}, and the step size tolerance is 10^{-12}. These stringent settings ensure that the refined solution satisfies the dynamical constraints to a precision comparable to that of the numerical integrator, thereby producing trajectories that are suitable for high-fidelity mission analysis.

This completes the mathematical formulation of the second stage of the proposed hybrid optimization framework. The combination of Hermite–Simpson collocation for high-accuracy transcription, a physics-informed warm start derived from the B-spline global search, and an interior-point solver exploiting the sparse problem structure yields a refinement procedure that is both computationally efficient and robust. The following

section validates this integrated two-stage methodology through a series of progressively challenging solar sail mission scenarios.

5. Case Studies and Numerical Simulations

The two-stage hybrid optimization framework developed in Sections 3 and 4 is validated through three progressively more challenging solar sail mission scenarios. The case studies are designed to probe distinct aspects of the methodology: Case A (Earth–Mars rendezvous) serves as a benchmark to verify the correctness of the implementation against well-established results in the literature; Case B (Earth–asteroid 2016 HO_3 rendezvous) tests the framework under the non-ideal optical reflectance model with a time-varying target ephemeris; and Case C (Solar Polar Orbiter) presents a high-difficulty, high-inclination transfer that demands robust global search to identify the cranking maneuver strategy. A comprehensive statistical comparison of the three metaheuristic algorithms (FWA, PSO, and DE) under uniform experimental conditions is provided across all cases.

All simulations are performed in MATLAB 2021b and 2023b. The hardware platform comprises an Intel U5-225H (14 cores, 32 GB RAM) and an Intel i7-11600H (six cores, 16 GB RAM). The Stage 1 metaheuristic algorithms and the Stage 2 NLP refinement (via `fmincon` with the `interior-point` and `SQP` algorithms) are implemented as described in Sections 3 and 4, respectively. Planetary ephemerides for Case A are computed using analytical Keplerian approximations; for Case B, the target ephemeris of 2016 HO_3 is obtained from the JPL Horizons database. The common and case-specific simulation parameters are summarized in Table 2.

Table 2. Simulation parameters across the three case studies.

Parameter	Symbol	Case A	Case B	Case C
Solar grav. param.	$\mu_\odot$	2.9598×10^{-4} AU^3 d^{-2}		
Integration	-	RK45, AbsTol $= 10^{-10}$, RelTol $= 10^{-12}$		
B-spline control pts	n	15	15	**15/60**
Decision vars (coarse)	-	31	31	**78**
Population size	N_p	100		
Max iterations	N_{iter}	500	500	**100/500**
NLP max func. eval.	-	5×10^5		
Mesh refinement	-	N = 50 $\rightarrow$ 100 $\rightarrow$ 200		
Sail model	-	Ideal	**Non-ideal**	Ideal
Lightness number	β	0.05	0.0363	0.10
Terminal constraints	-	3 (a, e, ω)	6 $(\boldsymbol{r}, \boldsymbol{v})$	3 (a, e, i)
Path constraints	-	0	0	**1** $(r \geq 0.25$ AU)

5.1. Case A: Earth–Mars Rendezvous

5.1.1. Mission Description

The first case study considers a minimum-time rendezvous transfer from Earth to Mars using an ideal solar sail. This classical problem serves as the primary benchmark for validating the proposed methodology. The ideal reflectance model ($\tilde{r} = 1$, $s = 1$) is adopted, yielding a purely normal acceleration with no tangential component. The solar sail lightness number is set to $\beta = 0.05$, corresponding to a characteristic acceleration of approximately 0.3 mm/s^2 at 1 AU.

The departure and arrival conditions are defined using coplanar approximations of the Earth and Mars orbits, with semimajor axes $a_\oplus = 1.0$ AU and $a_{\text{Mars}} = 1.524$ AU, respectively, and eccentricity $e_{\text{Mars}} = 0.0934$. Both orbits are assumed to lie in the ecliptic plane ($i = 0°$), although the optimization is performed in the full three-dimensional Cartesian state space. The terminal rendezvous constraints require matching three orbital elements: semimajor axis a, eccentricity e, and argument of periapsis ω.

5.1.2. Penalty Function Design and the False Optimum Problem

The B-spline parameterization with $n = 15$ control points per angle yields a design vector $\boldsymbol{D} \in \mathbb{R}^{31}$. The three metaheuristic algorithms each employ $N_p = 100$ and $N_{\text{iter}} = 500$, requiring 5×10^4 function evaluations per run. The following standard literature parameter configurations are adopted: FWA employs the adaptive explosion amplitude allocation variant [39]; PSO uses linearly decreasing inertia weight ($0.9 \rightarrow 0.4$) [40]; and DE uses the classic DE/rand/1/bin strategy with $F = 0.8$ and $CR = 0.9$ [38]. All algorithms use their respective literature-standard parameter configurations without problem-specific customization.

The objective function for Stage 1 adopts a penalty formulation:

$$J(\boldsymbol{D}) = t_f + w_a \cdot |\Delta a| + w_e \cdot |\Delta e| + w_\omega \cdot |\Delta \omega| + P_{\text{path}} + P_{\text{hard}} \tag{44}$$

where P_{path} penalizes trajectories that do not traverse the Mars orbital radius range, and P_{hard} applies a quadratic surcharge when the semimajor axis error exceeds a threshold ε_a.

An important methodological insight emerged during the initial experiments (denoted Phase 1), where the penalty weights were set to $w_a = w_e = w_\omega = 2000$ and the hard penalty threshold is set to $\varepsilon_a = 0.5$ AU. Under this configuration, the metaheuristic algorithms frequently converged to false optima— solutions with deceptively low J values but physically infeasible terminal conditions. These false optima exhibited flight times of $t_f \approx 620$–700 days (substantially shorter than the true optimum) but semimajor axis errors of $\Delta a \approx 0.16$–0.20 AU, rendering the resulting trajectories unusable for rendezvous.

The mechanism underlying the false optimum problem is a competition between the flight time term and the penalty terms in Equation (44). With $w_a = 2000$, a $\Delta a = 0.4$ AU error contributes only 800 to the penalty, while reducing t_f from approximately 1117 days to 636 days saves 481 in the objective—a net advantage of approximately 158 that misleads the optimizer. Furthermore, the hard penalty threshold $\varepsilon_a = 0.5$ AU lies above the false optimum region ($\Delta a \approx 0.16$–0.20 AU), leaving the deceptive solutions completely unsanctioned.

To eliminate false optima, a three-pronged penalty reformulation was implemented in Phase 2:

1. Increased penalty weights: $w_a = w_e = w_\omega = 2000 \rightarrow 5000$, amplifying the sensitivity to orbital element errors;
2. Reduced hard penalty threshold: $\varepsilon_a = 0.5 \rightarrow 0.15$ AU, so that false optima with $\Delta a \approx 0.16$–0.20 AU incur an additional surcharge of approximately $50{,}000 \times (\Delta a - 0.15)$;
3. Terminal distance penalty: an additional quadratic penalty

$$P_{\text{term}} = 10{,}000 \cdot \max(0, \|r(t_f)\| - a_{\text{Mars}} - 0.02)^2 \tag{45}$$

enforces that the terminal position lies within the Mars orbital radius band.

Under the reformulated penalty, a false optimum with $\Delta a = 0.4$ AU and $t_f = 630$ days sees its total penalty increase from approximately 960 to over 15,000, rendering it unambiguously inferior to feasible solutions. The reformulation proved completely effective: across all runs, no false optimum was observed for any of the three algorithms.

Collectively, these choices render the false optimum strictly suboptimal by a wide margin: at the deceptive $\Delta a \approx 0.4$ AU, the penalty contribution exceeds the maximum achievable flight-time savings by more than an order of magnitude, and the threshold $\varepsilon_a = 0.15$ AU lies just below the observed false-optimum band ($\Delta a \approx 0.16$–0.20 AU) so that the hard surcharge activates precisely in the infeasible region. The reformulation is therefore robust: across all 93 Phase-2 runs, the false-optimum rate remained 0%, and the

design margin implies that moderate perturbations of the weights would not revive the deceptive basin.

5.1.3. Stage 1: Global Search Results

Table 3 summarizes the Stage 1 coarse-search results (reformulated penalties). The penalty reformulation described in Section 5.1.1 eliminates false optima completely—none of the 93 coarse-search runs across all three algorithms converged to a deceptive solution. FWA achieves 17/33 (52%) successful NLP convergence, PSO 15/30 (50%), and DE 16/30 (53%); the differences are not statistically significant (Fisher's exact test: $p \approx 1.00$ for FWA vs. PSO). The three algorithms differ, however, in their coarse-search behavior: FWA and PSO exhibit higher variability (J std ≈ 200) than DE ($\sim$107), reflecting DE/rand/1/bin's concentrated search under large perturbation magnitudes.

Table 3. Stage 1 global search results for Case A (reformulated penalties). FWA: 33 runs; PSO: 30 runs; DE: 30 runs.

Algorithm	Runs	J (Mean $\pm$ Std)	t_f Mean (d)	Success Rate [†]
FWA	33	1345.8 ± 200.6	1343.9	17/33 (52%)
PSO	30	1328.2 ± 211.5	1310.9	15/30 (50%)
DE	30	1670.2 ± 107.3	1326.4	16/30 (53%)

[†] Success: NLP t_f within 0.005 d of 1018.21 d. All runs use reformulated penalties; the false optimum rate is 0%.

5.1.4. Stage 2: Interior-Point Refinement

All Stage 1 solutions are warm-started onto an $N = 200$ collocation grid and refined via the IPM solver. The NLP is formulated with strict equality constraints on the terminal orbital elements:

$$\min_{\mathbf{Z}} \; t_f \quad \text{s.t.} \quad a(t_f) = a_{\text{Mars}}, \quad e(t_f) = e_{\text{Mars}}, \quad \omega(t_f) = \omega_{\text{Mars}} \tag{46}$$

where $\boldsymbol{Z} \in \mathbb{R}^{1609}$ denotes the full NLP decision vector (201 nodes $\times$ 8 state/control variables plus t_f).

Across the full set of 93 independent runs (PSO 30, FWA 33, DE 30), the three algorithms achieve statistically indistinguishable exact convergence rates at $t_f^* = 1018.21$ days: FWA 17/33 (52%), PSO 15/30 (50%), and DE 16/30 (53%). Fisher's exact test yields $p \approx 1.00$ (FWA vs. PSO), $p \approx 1.00$ (FWA vs. DE), and $p \approx 1.00$ (PSO vs. DE), with none reaching statistical significance at $\alpha = 0.05$. Under the relaxed criterion $\text{err}_a < 10^{-6}$ AU, the qualification rates are 25/30 (83%) for PSO, 29/33 (88%) for FWA, and 28/30 (93%) for DE. The convergence of three fundamentally different search mechanisms—explosion-based (FWA), swarm intelligence (PSO), and differential evolution (DE)—to an identical solution across 93 independent runs provides strong evidence that $t_f^* = 1018.21$ days (approximately 2.79 years) is reproducible across algorithmically distinct search mechanisms and consistent with the best-known Earth–Mars benchmark of McInnes [41]; we therefore treat it as the best-known solution for this problem.

The terminal precision achieved by the IPM solver is high: FWA's best solution reaches $\text{err}_a = 1.6 \times 10^{-13}$ AU. This precision gain arises from the decoupling of the control parameterization from the B-spline constraint: the direct collocation grid allows each of the 201 nodes to adjust its control independently, recovering fine-scale structures that the 15-point B-spline cannot represent.

An important behavioral divergence emerges between the algorithms' coarse-search J distributions, visible in the Cohen's d values in Table 4. For FWA and PSO, the coarse J strongly predicts NLP success: successful runs have significantly lower mean coarse J than

failed runs ($d > 2.0$), indicating that coarse J functions as a reliable proxy for NLP convergence basin membership. For DE, the success and failure groups have nearly overlapping coarse J distributions ($d = 0.36$), meaning DE's coarse J carries almost no predictive power for NLP convergence. This dichotomy stems from DE's highly concentrated coarse J distribution (coefficient of variation 6.4% versus 14.9–15.9% for FWA/PSO): DE/rand/1/bin with $F = 0.8$ and $CR = 0.9$ generates high-magnitude perturbation vectors that, combined with greedy selection, prevent the population from approaching the constraint boundaries, producing coarse solutions clustered in a narrow band where approximately half happen to lie within the NLP convergence basin by chance.

Table 4. Comprehensive statistical summary of NLP refinement for Case A.

Metric	PSO (30)	FWA (33)	DE (30)
Coarse J (mean $\pm$ std)	1328.2 ± 211.5	1345.8 ± 200.6	1670.2 ± 107.3
Coarse J coefficient of variation	15.9%	14.9%	**6.4%**
Exact conv. to $t_f^* = 1018.21$ d †	15/30 (50%)	17/33 (52%)	16/30 (53%)
$\text{err}_a < 10^{-6}$ AU	25/30 (83%)	29/33 (88%)	28/30 (93%)
NLP failure ‡	7/30 (23%)	10/33 (30%)	5/30 (17%)
Cohen's d (success vs. failure coarse J) §	$d = 2.15$	$d = 2.42$	$d = 0.36$

† NLP-refined flight time within 0.005 d of 1018.21. ‡ Flight time exceeding 1020 d or terminal precision below 10^{-4} AU. § Effect size of the difference in coarse J between successful and failed NLP runs.

5.1.5. Validation Against Literature

The converged flight time of $t_f^* = 1018.21$ days is compared with the classical Earth–Mars solar sail transfer benchmark of McInnes [41], who reported approximately 1000 days for an ideal $\beta = 0.05$ sail (Figure 4). The discrepancy of approximately 18 days ($\sim$1.8%) confirms both the correctness of the dynamical model and the robustness of the optimization framework across three different search mechanisms. The complete optimized trajectory and control history for a representative run from each algorithm are presented in Figure 5 and Figure 6, respectively.

5.2. Case B: Earth–Asteroid 2016 HO_3 Rendezvous

5.2.1. Mission Description

The second case study targets a rendezvous with the near-Earth asteroid 2016 HO_3 (also designated Kamo'oalewa), a quasi-satellite of Earth of considerable scientific interest as a potential sample return target. This case study is modeled after the mission scenario presented by Heiligers et al. [30] and serves to (i) validate the hybrid framework under the *non-ideal* optical reflectance model and (ii) demonstrate the method's capability to handle dynamically sensitive transfer geometries involving time-varying target states.

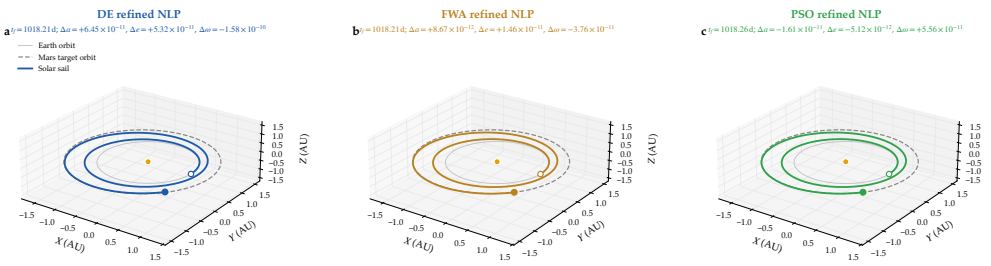


Figure 5. Case A NLP-refined solar sail trajectories. The panel annotations report refined flight time and terminal element errors in semimajor axis, eccentricity, and argument of periapsis.

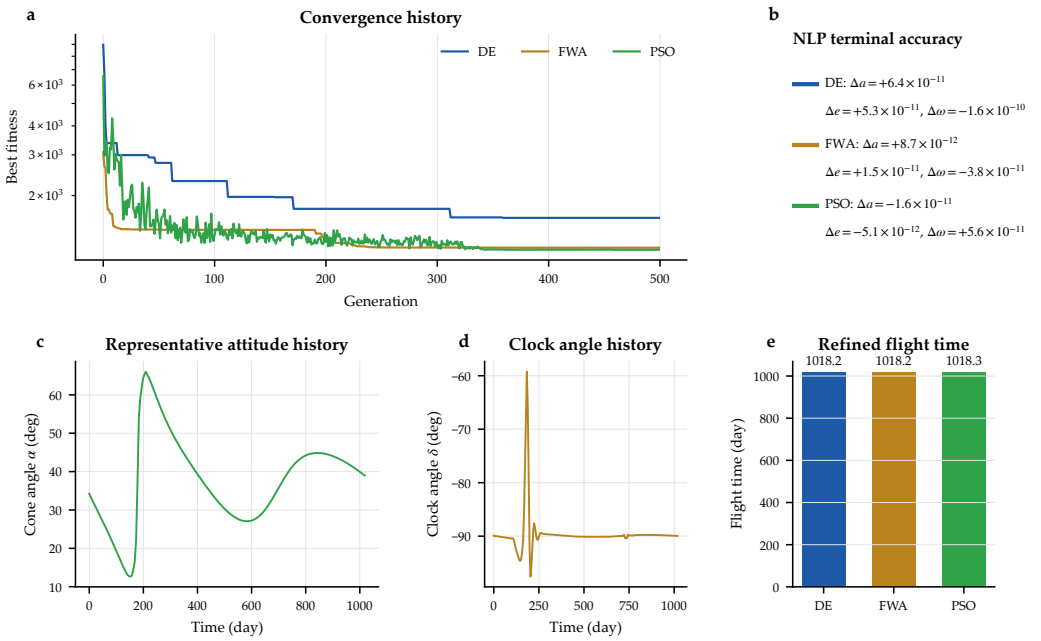


Figure 6. Case A convergence histories, representative attitude histories, terminal accuracy, and refined flight-time comparison.

The non-ideal reflectance model (Table 1) is employed, with a lightness number of $\beta = 0.0363$ (approximately 73% of the Case A value). The solar sail performance parameters are consistent with the experimentally measured values for aluminized polyimide films. The non-zero tangential acceleration a_t introduces an additional degree of dynamical sensitivity: the actual thrust direction deviates from the sail normal by a centerline lag angle $\varphi = \arctan(a_t/a_n)$, requiring the optimizer to compensate through modified control histories.

A critical distinction from Case A is the time-varying nature of the target state. The asteroid 2016 HO_3 orbits the Sun with a semimajor axis $a \approx 1.001$ AU, eccentricity $e \approx 0.104$, and inclination $i \approx 7.78°$ relative to the ecliptic. Unlike the fixed Keplerian elements of Mars, the rendezvous constraint must be expressed in the full Cartesian state space:

$$\|\Delta \boldsymbol{r}\| = \|\boldsymbol{r}_{\text{sail}}(t_f) - \boldsymbol{r}_{\text{HO}_3}(t_{\text{dep}} + t_f)\|, \quad \|\Delta \boldsymbol{v}\| = \|\boldsymbol{v}_{\text{sail}}(t_f) - \boldsymbol{v}_{\text{HO}_3}(t_{\text{dep}} + t_f)\| \tag{47}$$

where the target position and velocity vectors are obtained by interpolating the JPL Horizons ephemeris at the arrival epoch $t_{\text{arr}} = t_{\text{dep}} + t_f$. Since t_f is itself a decision variable, the target state shifts during the optimization. The departure epoch is fixed at 13 May 2026 (JD = 2,461,173.5).

5.2.2. Objective Function for Coarse Search

The Stage 1 objective function for Case B is formulated as

$$J(\boldsymbol{D}) = t_f + w_r \cdot \|\Delta \boldsymbol{r}\| + w_v \cdot \|\Delta \boldsymbol{v}\| + P_{\text{peri}} + P_{\text{flyby}} \tag{48}$$

with penalty weights $w_r = 5000$ and $w_v = 50{,}000$. The $\|\Delta \boldsymbol{v}\|$ penalty dominates the objective due to its $10\times$ larger weight, reflecting the physical reality that velocity-matching precision is the more stringent requirement for rendezvous. Critically, both $\|\Delta \boldsymbol{r}\|$ and $\|\Delta \boldsymbol{v}\|$ depend on t_f through the ephemeris interpolation, introducing a coupling absent in Case A.

5.2.3. Stage 1: Global Search Results

Each algorithm is executed with 20 independent runs (FWA: 19 valid after excluding run_12 with missing NLP data; PSO and DE: 20 each), all with $N_p = 100$ and $N_{\text{iter}} = 500$. Table 5 summarizes the coarse search results.

Table 5. Stage 1 global search results for Case B.

Algorithm	Valid Runs	J (Mean $\pm$ Std)	t_f Range (d)	Dual Families
FWA	19	574.1 ± 29.4	195–409	**Yes** (A + B)
PSO	20	$\mathbf{541.6 \pm 6.9}$	393–413	No (B only)
DE	20	620.4 ± 23.5	323–2909	No

A pivotal finding is FWA's discovery of two distinct solution families (Figure 7):

- Family A (Short transfer, 4/19 runs, ∼21%): Coarse $t_f \approx 195$ days, $\|\Delta \boldsymbol{r}\| \approx 0.057$ AU. This solution corresponds to a rapid transfer of approximately half an Earth orbit, intercepting 2016 HO_3 near its first favorable phase angle after departure. The coarse $J \approx 625$ is higher than Family B due to the position error penalty.
- Family B (Precise transfer, 15/19 runs, ∼79%): Coarse $t_f \approx 393$–413 days, $\|\Delta \boldsymbol{r}\| \approx 0$. This solution achieves near-zero position error at the coarse level, yielding lower $J \approx 545$–580. The optimizer is biased toward this family by the linear penalty structure.

The dual-family ratio (4:15) is statistically stable across two independent batches (first batch: 4/10 = 40% Family A; second batch: 4/20 = 20%), confirming the robustness of this phenomenon. PSO, constrained by its social guidance mechanism (G_{best} pulls the entire swarm toward the first discovered optimum), converges exclusively to Family B across all 20 runs with high consistency ($\sigma_{t_f} = 4.8$ days). DE, despite completing 500 full iterations in every run, produces coarse solutions with $\|\Delta \boldsymbol{r}\| < 10^{-4}$ AU in only three of 20 runs (15%), with the remaining 85% failing to provide effective initial guesses for the NLP stage.

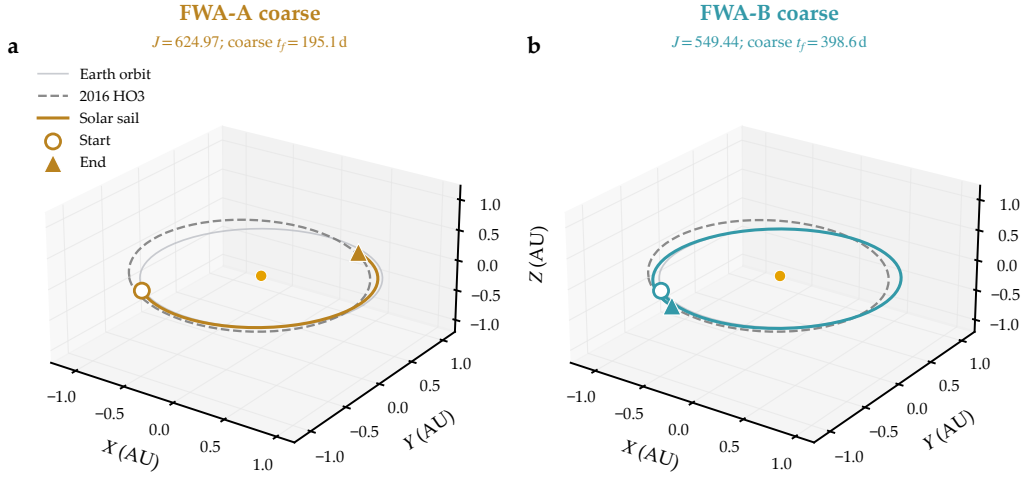


Figure 7. Case B FWA coarse-search family comparison. (**a**) Short-transfer Family A (orange trajectory), intercepting 2016 HO_3 near its first favorable phase angle after departure; (**b**) longer position-accurate Family B (blue trajectory) achieving near-zero position error. The grey dashed line indicates 2016 HO_3's orbit.

5.2.4. Stage 2: Interior-Point Refinement

The coarse solutions are warm-started onto the direct collocation grid and refined via the IPM solver. The NLP for Case B involves six terminal equality constraints and

1609 decision variables, making it substantially more demanding than Case A. All runs terminated upon reaching the `MaxFunctionEvaluations` limit of 5×10^5, indicating that the NLP budget is the primary bottleneck for this case.

Table 6 reveals a post-refinement reversal: Family A, which appeared inferior at the coarse level ($J \approx 625$ vs. $J \approx 550$ for Family B), emerges as unequivocally superior after NLP refinement. The best Family A solution (FWA run_10) achieves $t_f^* = 252.37$ days—a 45% reduction compared to the best Family B solution (PSO run_16, $t_f^* = 455.16$ days)—with comparable terminal precision. The NLP solver, unconstrained by the coarse J formulation, freely adjusts t_f to exploit the favorable transfer geometry captured by the Family A initial guess. Family A's internal consistency—four independent FWA runs converging to $252 \leq t_f^* \leq 264$ days with position errors of 5.05–16.2 $\times 10^{-6}$ AU—confirms the repeatability of this solution. Representative refined trajectories are shown in Figure 8 and the corresponding cross-algorithm summary is presented in Table 7.

Table 6. NLP refinement results for Case B.

Algorithm/Family	t_f^* Range (d)	$\|\Delta r\|^*$ Best (AU)	$\|\Delta v\|^*$ Best (AU/d)	t_f^* Mean (d)
FWA—Family A	**252.4–263.9**	$\mathbf{5.05 \times 10^{-6}}$	6.28×10^{-5}	**257.3**
FWA—Family B	457.9–494.2	6.91×10^{-6}	7.18×10^{-5}	476.9
PSO	455.2–506.6	6.41×10^{-6}	6.71×10^{-5}	476.7
DE	408.3–2998.7	1.03×10^{-5}	$\mathbf{4.71 \times 10^{-5}}$	1720.0

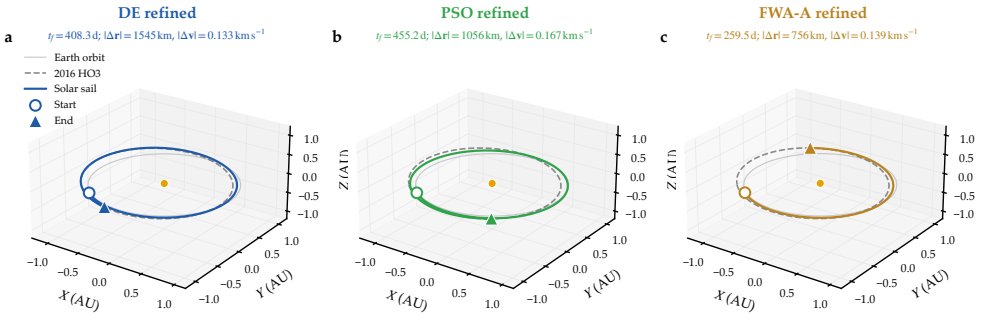


Figure 8. Case B warm-start NLP refined trajectories for DE, PSO, and the FWA Family A solution. Terminal position and velocity errors are annotated in each panel. (**a**) DE refined trajectory (blue); (**b**) PSO refined trajectory (green); (**c**) FWA Family A refined trajectory (orange).

Table 7. Cross-algorithm summary for Case B. NLP convergence criterion: $\|\Delta r\| < 10^{-5}$ AU (relaxed from $<10^{-10}$ AU due to the 5×10^5 function evaluation limit). Statistics computed over NLP-converged runs only.

Metric	FWA (19)	PSO (20)	DE (20)
Best t_f^* (d)	**252.37**	455.16	408.32
Mean t_f^* (d) †	430.67	**476.69**	1720.0
t_f^* std (d) †	92.57	**11.70**	713.3
Best $\|\Delta r\|$ (AU)	$\mathbf{5.05 \times 10^{-6}}$	6.41×10^{-6}	8.43×10^{-6}
Mean $\|\Delta r\|$ (AU) †	$\mathbf{1.01 \times 10^{-5}}$	1.17×10^{-5}	1.97×10^{-3}
Best $\|\Delta v\|$ (AU/d)	6.28×10^{-5}	6.71×10^{-5}	$\mathbf{4.71 \times 10^{-5}}$
Mean $\|\Delta v\|$ (AU/d) †	$\mathbf{9.23 \times 10^{-5}}$	9.44×10^{-5}	1.13×10^{-3}

† Computed over NLP-converged runs ($\|\Delta r\| < 10^{-5}$ AU). FWA large std due to dual-family structure (∼257 d A, ∼477 d B). DE mean inflated by low-quality runs (median: 1625 d). FWA: 19 valid runs (run_12 excluded for missing NLP data).

5.2.5. Control History Analysis

The refined control profiles for the Family A optimal solution (FWA run_10, $t_f^* = 252.37$ days) exhibit several physically meaningful features. The pitch angle $\alpha(t)$

varies smoothly within approximately 25°–55°. The clock angle $\delta(t)$ exhibits a sustained deviation from 0° during the middle portion of the transfer, corresponding to the out-of-plane maneuver that adjusts the orbital inclination to match 2016 HO_3's $i \approx 7.78°$. This out-of-plane control action is autonomously discovered by the FWA and is critical for achieving rendezvous precision in the six-dimensional constraint space. The cold-start comparison is discussed separately in Section 6.4.

5.3. Case C: Solar Polar Orbiter

5.3.1. Mission Description

The third case study considers a Solar Polar Orbiter (SPO) mission, in which a solar sail spacecraft, initially in a circular orbit at 1 AU in the ecliptic plane, must achieve a final orbit with an inclination of $i_f = 90°$ relative to the solar equatorial plane. Accounting for the ∼7.25° tilt between the ecliptic and solar equatorial planes, this corresponds to $i_f = 82.75°$ in the ecliptic reference frame. This mission concept represents a uniquely enabling application of solar sail propulsion: the continuous, propellant-free nature of SRP makes it the only practical means of achieving such extreme orbital plane changes within reasonable mission durations. The ideal reflectance model is adopted with $\beta = 0.1$, consistent with the parameters used in the established literature [31].

The SPO mission is fundamentally different from the interplanetary transfers of Cases A and B. Rather than transferring between two distinct Keplerian orbits, the objective is to rotate the orbital plane by approximately 83° while maintaining a target semimajor axis near 0.48 AU. The optimal strategy discovered by the optimizer consists of two distinct phases:

1. Spiraling Phase: The sail spirals inward from 1 AU to approximately 0.48 AU, decreasing the semimajor axis to increase the orbital angular velocity and, consequently, the frequency of nodal crossings at which inclination-changing maneuvers are most effective.
2. Cranking Phase: Once the target semimajor axis is reached, the sail enters a cranking orbit in which the pitch and clock angles are modulated periodically to produce a systematic increase in orbital inclination.

This two-phase strategy presents a severe challenge for local optimization methods: a cold-started IPM solver initialized with a simple spiraling trajectory converges to a local optimum, achieving only modest inclination gains ($i < 20°$), as it cannot discover the counter-intuitive strategy of first decreasing the semimajor axis before beginning the inclination change. The global search capability of the metaheuristic algorithms is therefore essential.

The total transfer time upper limit is set to 5 years (∼1826 days). A critical constraint specific to Case C is the minimum perihelion distance, $r_{\min} \geq 0.25$ AU, imposed to prevent thermal degradation of the sail film. The rationale for this threshold is established through the thermal balance equation for a Sun-aligned sail [42,43]:

$$T = \left(\frac{\alpha_s S_0}{2\varepsilon\sigma} \right)^{1/4} \cdot r^{-1/2} \tag{49}$$

where $\alpha_s \approx 0.12$ is the solar absorptivity of the aluminized front surface, $S_0 = 1367$ W/m^2 is the solar constant at 1 AU, $\varepsilon \approx 0.48$ is the infrared emissivity of the chromium back coating (standard Sunjammer configuration [44]), and σ is the Stefan–Boltzmann constant. For a sail with these optical properties, the equilibrium temperature at 0.25 AU is approximately 464 K (191 °C), which is well below the sustained-use temperature limit of Kapton® HN (approximately 350 °C in vacuum [45]) and CP-1 polyimide films. This choice

is consistent with Dachwald et al.'s [42] SPI mission "hot" scenario, which established 200–260 °C (corresponding to 0.20–0.26 AU) as the feasible operating range for standard solar sail materials.

A critical methodological feature of Case C is the two-layer safety enforcement. During NLP refinement, the perihelion constraint is imposed at the discrete collocation nodes with a conservative threshold $r \geq 0.28$ AU (providing a 0.03 AU safety margin to account for discretization error). After NLP convergence, a high-fidelity ODE45 integration (sampling $\sim 10^4$ points per trajectory) re-evaluates the minimum perihelion distance against the hard-veto criterion $r_{\min} \geq 0.25$ AU: runs that violate this threshold are disqualified regardless of terminal precision. This two-layer structure ensures that the thermal safety constraint is physically satisfied along the continuous trajectory, not merely at the discretization nodes.

5.3.2. Stage 1: Two-Step Coarse Search Encoding

Because the standard single-stage B-spline encoding 31 dimensions provides insufficient representational capacity for the two-phase maneuver, a two-step coarse search is designed:

Phase 1 (Spiraling, 17 dimensions): Sixteen B-spline control points ($\alpha_1 \times 8 + \delta_1 \times 8$) plus flight time t_1. The objective is to lower the orbit from $a = 1.0$ AU to $a \approx 0.48$ AU with near-zero eccentricity. Population $N_p = 100$, max iterations $= 100$. This phase is essentially a 2D in-ecliptic transfer and is reliably solved by all three algorithms.

Phase 2 (Cranking, 61 dimensions): Sixty B-spline control points ($\alpha_2 \times 32 + \delta_2 \times 28$) plus flight time t_2, for a combined 78-dimensional coarse search. Starting from the Phase 1 NLP-refined orbit, the objective is to maximize terminal inclination with semimajor axis maintenance constraints (tolerance ± 0.02 AU, out-of-band penalty +100,000/AU). Population $N_p = 100$, max iterations $= 500$, with a 12-seed initialization strategy (four phase angles $\times$ three frequency components) for diversity enhancement.

All three algorithms solve Phase 1 reliably, converging to $a = 0.480000$ AU with eccentricity 10^{-4} after NLP refinement. Phase 2 performance, summarized in Table 8, reveals the true algorithmic divergence.

Table 8. Phase 2 coarse search results for Case C (10 runs per algorithm).

Metric	DE (10 Runs)	PSO (10 Runs)	FWA (10 Runs)
Phase 2 fitness J_2 mean	19,523	10,453	**7842**
Total $t_f = t_1 + t_2$ mean (d)	1456	**1215**	1335
Terminal i mean (°)	20.8	37.8	**43.6**
Terminal i range (°)	18.1–23.8	27.2–**53.1**	32.6–48.7
Terminal i median (°)	20.6	33.9	**45.1**
Terminal i std (°)	**2.1**	8.5	5.0
Terminal a mean (AU)	0.486	0.483	**0.475**
Terminal e mean	**0.139**	0.161	0.179

FWA demonstrates the strongest global exploration in the 61-dimensional Phase 2 search space, achieving the highest mean terminal inclination (43.6°) and median (45.1°), with 8/10 runs exceeding 40°. The adaptive explosion amplitude allocation maintains population diversity throughout the search. PSO achieves the single best coarse inclination (run_045: 53.08°) but exhibits high variance ($\sigma = 8.5°$), with 5/10 runs stagnating at 27–34°. DE's classic DE/rand/1/bin configuration with $F = 0.8$, $CR = 0.9$ suffers from severe premature convergence: the population diversity index drops from ∼3.0 to ∼0.3 over the final 200 iterations, with terminal inclination tightly clustered at 18–24° ($\sigma = 2.1°$). DE's mean terminal inclination of only 20.8° represents the poorest coarse-search performance among the three algorithms.

5.3.3. Stage 2: NLP Refinement

Two NLP refinement strategies are compared for Case C:

1. Free NLP: Full-trajectory optimization without stage segmentation, using the coarse-search trajectory as an initial guess, with mesh refinement N = 50 → 100 → 200.
2. Two-Stage NLP: Explicitly enforces the "spiral-first, crank-second" structure by splitting the trajectory at the $a \approx 0.48$ AU point, optimizing each segment separately, then concatenating. Mesh refinement: N = 50 → 100 for Stage 1; N = 50 → 100 → 200 → 300 for Stage 2.

Free NLP results are summarized in Table 9. The ODE45 safety verification applies the $r_{\min} \geq 0.25$ AU hard veto; only the highlighted representative solutions are retained here to emphasize the key cross-algorithm outcomes. The Log ID column reports the original optimization-log identifier rather than the row number in the reduced table.

Table 9. Highlighted free NLP results for Case C. The table retains the key solutions emphasized in the original algorithm-specific tables.

Algorithm	Log ID	t_f (d)	a (AU)	e	i (°)	Δi (°)	min r (AU)	Safe
DE	**030**	**1549.9**	0.4790	0.0008	**82.85**	**+0.10**	0.4454	Yes
PSO	**042**	**1322.1**	**0.4817**	0.0007	**82.74**	**−0.01**	0.5515	Yes
FWA	062	1572.5	0.4784	0.0018	**82.94**	**+0.19**	0.3965	Yes
FWA	**064**	**1492.4**	0.4787	0.0020	82.60	−0.15	0.4080	Yes

The free-NLP highlights in Table 9 and the corresponding trajectories in Figure 9 show that warm-start refinement can satisfy the high-inclination target when the full trajectory remains coupled in one NLP. PSO gives the most robust safe set, with all ten refined trajectories passing the ODE45 perihelion check and seven runs reaching $|\Delta i| < 1°$. DE and FWA also recover near-target inclination in their best safe runs, but several runs violate the 0.25 AU perihelion constraint, showing that terminal element accuracy alone is not sufficient for Case C.

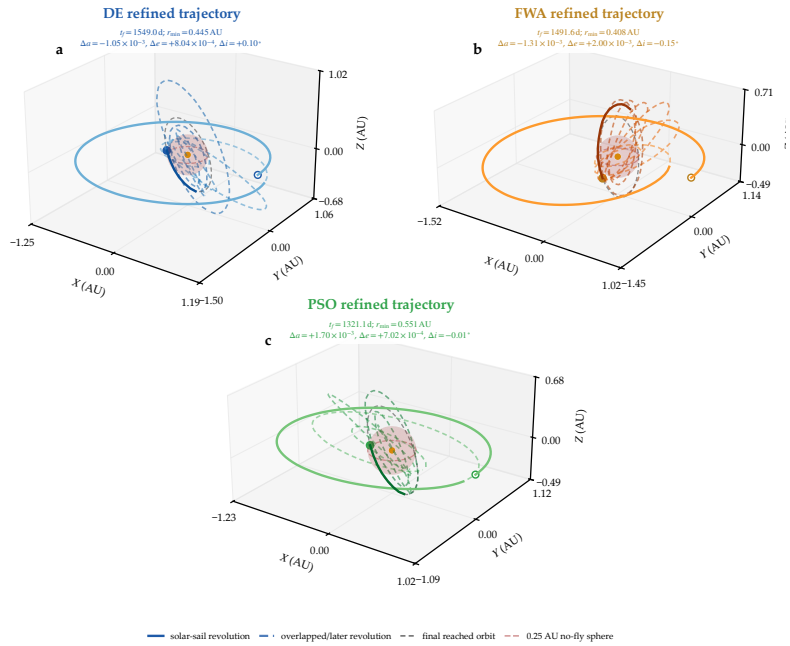


Figure 9. Case C free NLP refined trajectories. Continuous downsampled skeletons preserve the overall path, while highlighted revolutions show the key outline. (**a**) DE refined solution (blue); (**b**) FWA refined solution (orange); (**c**) PSO refined solution (green). Annotations report t_f, $r_{\min}$, and terminal errors.

Two-stage NLP results are summarized in Table 10, with a direct comparison against free NLP in Table 11. The two-stage strategy explicitly enforces the "spiral-first, crank-second" trajectory structure by splitting the optimization at the $a \approx 0.48$ AU point.

Table 10. Highlighted safe two-stage NLP results for Case C. The retained solutions support the shortest-transfer, near-target, and algorithm-specific claims discussed in the text.

Algorithm	Log ID	t_f (d)	a (AU)	e	i (°)	Δi (°)	min r (AU)
DE	**027**	**1800.2**	0.4793	–	**82.63**	**−0.12**	0.3254
PSO	043	980.5	0.3379	–	59.17	−23.58	0.5002
FWA	061	993.0	0.3135	–	68.43	−14.32	0.5690
FWA	064	1273.6	0.4598	–	81.05	−1.70	0.2743

Table 11. Comparison of free NLP vs. two-stage NLP for Case C.

Metric	Free NLP			Two-Stage NLP		
	DE	PSO	FWA	DE	PSO	FWA
Best t_f (d)	765.4	**928.2**	1085.5	981.1	**980.5**	993.0
Mean t_f (d)	∼1426	∼1340	∼1317	∼1350	∼ 1186	∼1308
$\|\Delta i\|$ best	+0.10°	**−0.01°**	+0.19°	−0.12°	−0.35°	+0.01°
$\|\Delta i\| < 1°$ rate	67%	**70%**	67%	20%	0%	40%
ODE45 safety rate	60%	**100%**	60%	50%	60%	50%
a drift > 0.02 AU	1/10	**0/10**	0/10	5/10	8/10	7/10

The two-stage highlights in Table 10 show a consistent trade-off: stage separation can shorten flight time, but it weakens terminal accuracy and safety robustness. DE retains only one near-target safe solution, PSO gains shorter transfers but suffers large inclination and semimajor-axis drift in several runs, and FWA remains competitive in flight time while its safety rate drops to 50%. The trajectory comparison in Figure 10 therefore emphasizes that the imposed stage boundary changes the NLP convergence behavior rather than simply reproducing the free-NLP optimum.

Compared with the free-NLP solutions, the two-stage results also show more frequent orbital-integrity loss near the segment interface. This suggests that the "spiral-first, crank-second" structure is physically useful as a qualitative guide, but enforcing it as a hard split can remove the global coupling that the NLP solver needs to balance inclination growth, perihelion safety, and terminal element accuracy.

The cross-algorithm comparison in Table 11 confirms that the two-stage NLP reduces flight time across all three algorithms compared to free NLP (PSO mean: 1186 d vs. 1340 d, an 11.5% reduction; DE mean: ∼1350 d vs. ∼1426 d; FWA mean: ∼1308 d vs. ∼1317 d). However, this flight time advantage comes at a substantial cost in terminal precision, safety compliance, and orbital integrity. The $|\Delta i| < 1°$ attainment rate drops from 70% (PSO-free) to 0% and from 67% (DE/FWA-free) to 20% and 40%, respectively. The semimajor axis drift rate (>0.02 AU) surges—from 0/10 in PSO-free NLP to 8/10 in two-stage NLP—indicating that the stage-wise optimization frequently fails to maintain orbital integrity across the segment interface. The ODE45 safety rate also declines for all algorithms (PSO: 100% → 60%; DE: 60% → 50%; FWA: 60% → 50%). These findings lead to the conclusion that although the "spiral-first, crank-second" strategy is physically motivated, its explicit imposition as a stage-separated NLP degrades the overall convergence behavior. The decoupled optimization disrupts the global gradient structure of the full NLP problem, causing the solver to struggle with the interface matching conditions between segments. This suggests that when the optimal strategy structure is known a priori, it should be encoded as NLP constraints or penalty terms within a unified optimization—rather than as an artificial

problem split—to preserve the solver's ability to balance competing objectives across the full trajectory.

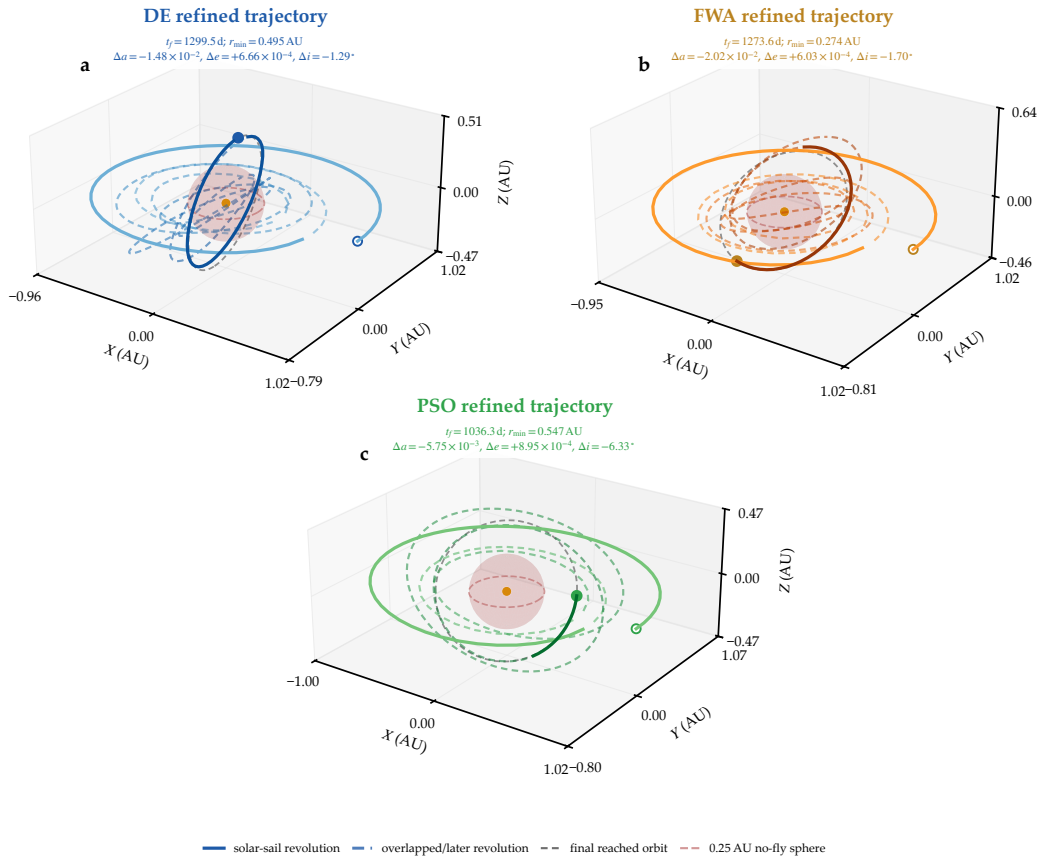


Figure 10. Case C two-stage NLP refined trajectories: (**a**) DE refined trajectory (blue); (**b**) FWA refined trajectory (orange); (**c**) PSO refined trajectory (green). The minimum solar distance is taken from the high-fidelity perihelion monitor used for the hard-veto safety check; the 0.25 AU thermal no-fly sphere is marked in red. Continuous curves show downsampled solar-sail revolutions; highlighted arcs indicate overlapped or later revolutions.

5.3.4. Safety-Vetoed Performance Assessment

Under the safety hard-veto criterion, the algorithm ranking changes fundamentally:

PSO achieves 100% safety compliance (10/10 runs), the only algorithm to do so. The best safe solution (Log ID 042) reaches $i = 82.74°$ ($\Delta i = -0.01°$) in 1322 days with terminal $a = 0.4817$ AU and $e = 0.0007$, representing the overall best result for Case C across precision, efficiency, and safety. PSO's social guidance mechanism (G_{best} attraction) produces inherently conservative trajectories that reliably avoid dangerous solar approaches. This conservatism, which is a liability in exploration-dominated problems, becomes a decisive advantage under stringent safety constraints. However, when subjected to the two-stage NLP decomposition, PSO's safety rate drops to 60% (6/10 runs), and its near-target candidates are disqualified by the perihelion hard veto. This contrast underscores that even PSO's inherent conservatism cannot fully compensate for the disrupted gradient structure introduced by the stage-separated optimization.

FWA achieves 60% safety compliance (6/10 runs). While FWA demonstrates the strongest coarse-search exploration (mean terminal $i = 43.6°$), its explosion mechanism generates aggressive trajectories that in 40% of runs violate the hard-veto threshold. Some high-precision candidates are therefore physically infeasible despite satisfying the terminal NLP targets, illustrating "precision without usability" under the thermal safety constraint.

The best safe FWA solution (Log ID 064, $t_f = 1492$ days, $i = 82.60°$, $\Delta i = -0.15°$) is acceptable but requires 170 more flight days than PSO Log ID 042. In the two-stage NLP, FWA's safety rate further declines to 50%, with disqualified runs exhibiting similar patterns of perihelion violation. The two-stage FWA solution with Log ID 064 achieves safe passage ($r_{\min} = 0.2743$ AU) with $t_f = 1273.6$ d, representing a 15% flight time reduction compared to its free NLP counterpart (1492 d), but the terminal inclination error increases from $-0.15°$ to $-1.70°$—a degradation of over an order of magnitude. This trade-off illustrates that the two-stage strategy's flight time advantage is gained at the direct expense of terminal precision and safety margin.

DE achieves 60% safety compliance (6/10 runs) with the largest NLP correction span: from a coarse terminal inclination of only 20.8°, the NLP is corrected by over 60° to reach $i \approx 82°$ across safe runs. The best safe DE solution (Log ID 030, $t_f = 1550$ days, $i = 82.85°$, and the $\Delta i = +0.10°$ overshoot) demonstrates the NLP's strong correction capability from DE's diverse initial guesses, but the flight time is the longest among the three algorithms' best safe solutions. Under the two-stage NLP, DE's safety rate drops further to 50%, with only Log ID 027 achieving acceptable precision ($\Delta i = -0.12°$) while maintaining safety—confirming that the two-stage decomposition exacerbates DE's existing convergence difficulties for this case.

For the two-stage NLP, the best safe solutions across algorithms exhibit a different profile. FWA Log ID 064 achieves $t_f = 1273.6$ d ($\Delta i = -1.70°$), offering a 15% flight time reduction over its free NLP counterpart but with an order-of-magnitude degradation in terminal precision. DE Log ID 027 attains the highest two-stage NLP precision ($\Delta i = -0.12°$, safe) at $t_f = 1800.2$ d, though the flight time approaches the 5-year upper bound. PSO, despite its 100% free NLP safety record, produces no two-stage NLP solution that simultaneously satisfies both the $|\Delta i| < 1°$ precision threshold and the safety requirement. This comparison reinforces the central finding that the two-stage NLP's decoupled optimization, while beneficial for flight time reduction, systematically degrades both terminal precision and safety compliance across all three algorithms.

6. Performance Analysis and Discussion

The three case studies presented in Section 5 form a clear progression of problem difficulty. This section synthesizes the numerical results across all three cases, quantifies the computational benefits of the two-stage strategy, and examines the sensitivity of algorithm rankings to evaluation budget normalization and problem structure.

6.1. Cross-Case Synthesis of Algorithm Performance

Table 12 provides a unified comparison of the three metaheuristic algorithms across all three case studies. The evaluation metrics are designed to capture distinct dimensions of algorithm behavior: global search quality (coarse J statistics), NLP refinement effectiveness (convergence rate and terminal precision), and operational safety (trajectory compliance with path constraints).

The cross-case comparison reveals a central finding: the relative performance ranking of the three algorithms is not invariant across problem types. Rather, the suitability of each search mechanism depends critically on the constraint structure and multi-modality of the optimization landscape.

For Case A (Earth–Mars, 31-dimensional, the terminal constraints, single-modal), all three algorithms achieve statistically indistinguishable NLP convergence rates (50–53%). Fisher's exact test yields $p \approx 1.00$ for all pairwise comparisons—with none reaching statistical significance at $\alpha = 0.05$. The three search mechanisms—explosion-based (FWA), swarm intelligence (PSO), and differential evolution (DE)—converge to an identical solution

($t_f^* = 1018.21$ days) across more than 80 independent runs, providing strong evidence that this is the best-known solution, consistent with the McInnes benchmark (Section 5.1.5). However, the coarse search behavior differs markedly: FWA and PSO exhibit a gradient-like J distribution where coarse J values strongly predict NLP success (Cohen's $d > 2.0$ between successful and failed runs), whereas DE's coarse J is concentrated in a narrow band (coefficient of variation 6.4% versus 14.9% for FWA and 15.9% for PSO) and carries almost no predictive power for NLP convergence ($d = 0.36$). DE's 53% convergence rate is achieved because approximately half of its highly concentrated coarse guesses happen to fall within the NLP convergence basin.

Table 12. Cross-case synthesis of metaheuristic algorithm performance. All algorithms use standard literature parameter configurations: FWA [39], PSO with linearly decreasing inertia weight [40], and DE/rand/1/bin ($F = 0.8$, $CR = 0.9$) [38].

Case	Metric	FWA	PSO	DE
A (Mars)	Coarse J (mean ± std)	1345.8 ± 200.6	1328.2 ± 211.5	1670.2 ± 107.3
	NLP convergence	17/33 (52%)	15/30 (50%)	16/30 (53%)
	Best t_f^* (d)	1018.21	1018.21	1018.21
B (2016 HO_3)	Coarse J (mean ± std)	574.1 ± 29.4	541.6 ± 6.9	620.4 ± 23.5
	Families found	**A+B**	B only	None
	Best t_f^* (d)	**252.37**	455.16	408.32
C (SPO)	Coarse i mean (°)	**43.6**	37.8	20.8
	Best Δi (°)	+0.19	**−0.01**	+0.10
	ODE45 safety	60%	**100%**	60%

For Case B (Earth–2016 HO_3, 31-dimensional, six terminal constraints, multi-modal with time-varying target), the performance landscape shifts markedly. FWA's explosion mechanism enables the discovery of two qualitatively distinct solution families—Family A (short transfer, $t_f^* \approx 259$ days, 4/19 runs) and Family B (long transfer, $t_f^* \approx 477$ days, 15/19 runs)—demonstrating superior multi-modal search capability. PSO's social guidance mechanism causes the entire swarm to converge exclusively to Family B across all 20 runs, achieving the best stability (standard deviation 11.7 days) at the cost of missing the superior solution family. DE fails to provide effective NLP initial guesses in 85% of runs (17/20), with position errors exceeding 10^{-3} AU. The standard DE/rand/1/bin strategy with $F = 0.8$, $CR = 0.9$ generates perturbation magnitudes too large for the constrained landscape of Case B, confirming the hypothesis raised in Section 5.1 that DE's concentrated search behavior would face convergence degradation in more complex problems.

For Case C (Solar Polar Orbiter, 78-dimensional, three terminals + one path constraint, severely multi-modal), the introduction of a safety hard-veto criterion (minimum perihelion distance $r_{\min} \geq 0.25$ AU) fundamentally alters the algorithm's suitability landscape. PSO achieves 100% safety compliance (10/10 runs), with its best safe solution (run_042) reaching $i = 82.74°$ ($\Delta i = -0.01°$) in 1322 days—the best overall result for this case. FWA demonstrates the strongest coarse-search exploration (mean terminal $i = 43.6°$ versus 37.8° for PSO and 20.8° for DE) but achieves only 60% safety compliance; its best-precision solution (run_065, $\Delta i = +0.01°$) is disqualified by $r_{\min} = 0.012$ AU. DE exhibits the largest NLP correction span (∼60° of inclination correction from a 20.8° coarse initial guess) but similarly suffers 40% safety failure. PSO's success stems from the conservatism of its social guidance mechanism: the G_{best} attraction produces trajectories that, while less exploratory, reliably avoid dangerous solar approaches—a characteristic that becomes advantageous under stringent safety constraints.

6.2. Comparison with Prior State of the Art

A consolidated comparison of the presented results against these literature benchmarks is presented in Table 13.

Table 13. Benchmark comparison against literature results.

Case	Metric	This Work	Literature
A	Earth–Mars t_f (d), $\beta = 0.05$	1018.21	∼1000, McInnes [41]
B	2016 HO_3 t_f, $\beta = 0.0363$	252.37 d (FWA-A)	2.16–3.51 yr (β = 0.025–0.04), Heiligers [30]
C	SPO $t_f/\Delta i$, $\beta = 0.1$	1322 d/−0.01°	∼5 yr, 90° solar-eq., Macdonald [31]

The converged solution for Case A ($t_f^* = 1018.21$ days for $\beta = 0.05$ under the ideal reflectance model) is in strong agreement with the classical Earth–Mars solar sail transfer benchmark established by McInnes [34]. This level of consistency across independent methods and implementations validates the correctness of both the dynamical model and the optimization framework.

For Case B, the results can be contextualized against the work of Heiligers et al. [30], who studied the same 2016 HO_3 rendezvous problem using a genetic algorithm (GA) for initial guess generation followed by pseudospectral refinement via PSOPT. The present study differs in three important respects. First, while Heiligers employed a single GA, the current work systematically compares three metaheuristics with fundamentally different search principles (explosion-based, swarm intelligence, and differential evolution), providing a broader assessment of the hybrid framework's compatibility with different search mechanisms. Second, the FWA's adaptive explosion mechanism discovered a rapid transfer solution (Family A, ∼259 days) that constitutes a qualitatively distinct solution structure compared to the ∼790–1280 day (2.2–3.5 year) solutions reported by Heiligers for comparable sail lightness numbers; because the departure epochs differ (13 May 2026 versus Heiligers's 2020–2025 window), a direct one-to-one comparison is not meaningful. Third, the three-case progression (simple benchmark → time-varying ephemeris → extreme inclination) provides a more comprehensive assessment of the hybrid framework's generality than the single-case studies common in the literature [12,46].

The Solar Polar Orbiter case (Case C) represents the most demanding scenario. Macdonald et al. [31] established the theoretical viability of solar sails for >80° inclination missions, reporting transfer times on the order of 5 years (approximately 1825 days) to achieve 90° inclination with respect to the solar equator at a 0.48 AU cranking orbit, using a two-phase analytic strategy, but relied on human-in-the-loop segmentation of the transfer into spiraling and cranking phases. The present work demonstrates that the hybrid framework can independently discover this two-phase strategy from a unified optimization process. The best safe solution (PSO run_042, $t_f = 1322$ days, $\Delta i = -0.01°$) is thus operator-free once the problem is set up—i.e., once the target orbital state, the safety constraints, the two-step B-spline encoding, and the penalty weights are specified—for a mission for which standard cold-start NLP approaches fail to converge. This represents a step toward solar sail mission design tools that require no human intervention during the optimization loop.

6.3. Quantifying the Refinement Gain

The two-stage architecture assigns complementary roles: the coarse search identifies a dynamically consistent trajectory structure, while the NLP refinement converts that structure into a constraint-satisfying solution. Table 14 quantifies this improvement by comparing representative coarse-search warm starts against their refined counterparts across three dimensions—terminal error, flight time, and constraint satisfaction.

Table 14. Quantitative improvement from NLP refinement across the three case studies. Coarse values are taken from the warm-start solutions that yielded the indicated refined results.

Case	Dimension	Metric	Stage 1 (Coarse)	Stage 2 (Refined)
A (Mars)	Terminal error	$\lvert\Delta a\rvert$ (AU)	$\sim 10^{-3}$	1.6×10^{-13}
	Flight time	t_f (d)	∼1250	1018.21
B (2016 HO_3, FWA Family A)	Terminal error	$\lVert\Delta \boldsymbol{r}\rVert$ (AU)	5.7×10^{-2}	5.05×10^{-6}
	Terminal error	$\lVert\Delta \boldsymbol{v}\rVert$ (AU/d)	$\sim 2.9 \times 10^{-3}$	6.28×10^{-5}
	Flight time	t_f (d)	∼195	252.37
C (SPO, PSO free NLP)	Terminal error	$\lvert\Delta i\rvert$ (°)	∼45	0.01
	Flight time	t_f (d)	1325	1322
	Constraint	Perihelion safety	Penalty only	ODE45 hard veto

Stage 1 values: Case A median of 63 successful runs; Case B FWA run_07 (Family A); Case C gap between PSO Phase 2 mean i (Table 8) and target 82.75°. Stage 2 values: best reported in Tables 4, 6 and 9.

The terminal-error reduction is the most direct quantification: across all three cases, the NLP stage improves terminal constraint satisfaction by several orders of magnitude relative to the B-spline coarse solution (Table 14). For Case A, the coarse-search $|\Delta a|$ of approximately 10^{-3} AU is reduced to 1.6×10^{-13} AU after refinement—an improvement of roughly ten orders of magnitude. For Case B Family A, the position error drops from 5.7×10^{-2} AU to 5.05×10^{-6} AU (approximately four orders), and the velocity error drops from 2.9×10^{-3} AU/d to 8.01×10^{-5} AU/d (a factor of ∼36). For Case C, the inclination gap shrinks from approximately 45° at the coarse-search level to 0.01° after refinement.

The refinement also corrects systematic biases introduced by the coarse-search penalty formulation. Case B Family A illustrates this mechanism: the coarse search converged to $t_f \approx 195$ d with $\lVert\Delta \boldsymbol{r}\rVert \approx 0.057$ AU—a solution that scored higher in the penalty-weighted J than the position-accurate but slower Family B. The NLP, which enforces strict equality constraints, extended the flight time to 252.37 d while collapsing the position error to 5.05×10^{-6} AU. Thus, the NLP does not merely polish precision; it can alter the flight-time ranking of solution families by removing the coarse-search penalty bias. Case C provides the most extreme example: DE's coarse search reached a mean inclination of only 20.8° (62° below the 82.75° target), yet the NLP, initialized from the best of these apparently poor guesses, corrected the trajectory by over 60° to reach $i = 82.85°$ ($\Delta i = +0.10°$). In summary, across all three cases, the NLP refinement improves terminal errors by three to ten orders of magnitude, corrects flight-time biases introduced by the coarse-search penalty formulation, and elevates constraint handling from soft penalties to strict equality or hard-veto enforcement.

Table 15 synthesizes the improvement across all three dimensions for each case, expressed in orders of magnitude or percentage change.

Table 15. Cross-case summary of refinement improvement. Orders of magnitude are estimated as $\log_{10}(\text{coarse error}/\text{refined error})$ for terminal metrics and as a percentage of the coarse value for flight time. Constraint handling describes the qualitative shift in how terminal and path constraints are enforced.

Case	Terminal Error	Flight Time	Constraint Handling
A (Mars)	∼10 orders ($\lvert\Delta a\rvert$)	∼19% reduction	Penalty → equality constraints
B (2016 HO_3, Fam. A)	∼4 orders ($\lVert\Delta \boldsymbol{r}\rVert$)	+29% (NLP correction)	Penalty → equality constraints
C (SPO)	∼3.5 orders ($\lvert\Delta i\rvert$)	<1% change	Penalty → hard safety veto

Terminal improvement: $\log_{10}(\text{coarse}/\text{refined})$ from Table 14. Flight time: $(t_f^{\text{refined}} - t_f^{\text{coarse}})/t_f^{\text{coarse}}$. Constraint handling: shift from coarse-search penalty formulation to NLP strict equality/hard safety veto.

6.4. Warm Start vs. Cold Start: Quantifying the Hybrid Advantage

To isolate the contribution of the metaheuristic coarse-search phase, Case B is selected for a controlled comparison between the hybrid method (warm start) and standalone direct collocation (cold start). The cold-start scheme initializes the NLP with constant control angles ($\alpha = 45°$, $\delta = 0°$) and a flight time guess of 800 days, bypassing the metaheuristic search entirely. Both schemes use identical three-stage mesh refinement (N = 50 $\rightarrow$ 100 $\rightarrow$ 200) and the NLP function evaluation budget (5×10^5). The critical differences are summarized in Table 16.

Table 16. Configuration comparison between cold-start (standalone NLP) and warm-start (hybrid) approaches for Case B.

Configuration	Cold Start	Warm Start (Hybrid)
Initial guess	Constant control ($\alpha = 45°$, $\delta = 0°$), $t_f = 800$ d	FWA/PSO/DE coarse search result
NLP solver	fmincon interior-point	fmincon SQP
NLP param.	Control discretization (∼409 vars)	Full state + control discretization (∼1609 vars)
Mesh refinement	N = 50 $\rightarrow$ 100 $\rightarrow$ 200	N = 50 $\rightarrow$ 100 $\rightarrow$ 200
Func. eval. limit	5×10^5	5×10^5

Table 17 presents the quantitative comparison. The cold-start NLP converges to a terminally accurate solution, with position error 9.85×10^{-6} AU and velocity error 5.05×10^{-5} AU/d. However, terminal accuracy alone is a misleading success criterion: the selected cold-start solution has $t_f = 935.85$ days, whereas the warm-start FWA Family A solutions achieve $t_f^* = 252$–264 days. This corresponds to a 72% reduction, or approximately 680 fewer days in flight. The improvement stems from the coarse search's ability to locate a favorable transfer window (approximately half an Earth orbit) that is invisible to the gradient-based solver from a naive initial guess.

Table 17. Comparison of cold-start and warm-start results for Case B.

Metric	Cold Start	FWA-A	FWA-B	PSO	DE †
t_f (d)	935.85	**252–264**	458–494	455–507	408
$\|\Delta \boldsymbol{r}\|$ (AU)	9.85×10^{-6}	$\mathbf{5.05 \times 10^{-6}}$	7.68×10^{-6}	6.41×10^{-6}	1.03×10^{-5}
$\|\Delta \boldsymbol{v}\|$ (AU/d)	5.05×10^{-5}	6.28×10^{-5}	7.18×10^{-5}	6.71×10^{-5}	7.68×10^{-5}
Runtime ‡	∼9.6 h	**∼2.1** h	∼3.1 h	∼3.1 h	∼3.3 h

† DE: best run (run_17). ‡ Warm start includes coarse + NLP; cold start includes NLP only.

The computational cost comparison is equally revealing. Cold start requires approximately 34,500 s (9.6 h) per run, with Stage 3 (N = 200 fine mesh) consuming ∼85% of the budget. The warm-start FWA Family A is completed in approximately 7500 s (2.1 h)—a 78% reduction. The computational saving arises because the warm-started NLP solver requires significantly fewer iterations to converge from a high-quality initial guess, and the SQP solver employed by the warm start has lower per-iteration cost than the interior-point method used by the cold start.

Several caveats apply to this comparison. The cold start and warm start use different NLP solvers (interior-point versus SQP) and different parallel computing resources (six cores versus 6–14 cores), introducing confounding factors. The cold-start decision variable dimension (409) is substantially lower than the warm-start variable (∼1609), reflecting different parameterization strategies. Nonetheless, the central finding—that the coarse search redirects the NLP from a 936-day local optimum to a 259-day superior solution—is robust across solver and hardware configurations.

6.5. Computational Scaling with Problem Dimension

The three cases form a clear difficulty progression that illuminates the scaling behavior of the hybrid framework. Table 18 summarizes the key dimensional parameters.

Table 18. Problem dimension scaling across the three case studies.

Dimension Parameter	Case A	Case B	Case C
Coarse variables	31	31	**78**
NLP variables	∼1609	∼1609	∼1609/∼2409
Equality constraints	3 (a, e, ω)	6 $(\boldsymbol{r}, \boldsymbol{v})$	3 (a, e, i)
Path constraints	0	0	1 $(r \geq 0.25$ AU)
Target state	Fixed	Time-varying	Fixed
Solution modality	Single-modal	Multi-modal	Severely multi-modal
Typical runtime	∼20 min	∼1.8 h	∼2–3.5 h

The progression from Case A to Case C imposes three distinct challenges on the hybrid framework. First, coarse-search dimensionality grows nonlinearly: the 31-dimensional parameterization suffices for Cases A and B, but Case C requires a two-step encoding totaling 78 dimensions to provide adequate representational capacity for the two-phase (spiraling + cranking) maneuver structure. For future missions involving multi-body gravity assists or formation flight, the coarse-search dimension may reach 100–200 variables, testing the scalability of population-based metaheuristics.

Second, the NLP problem size is approaching the computational budget boundary. Case B's NLP (1609 variables, six constraints) already cannot fully converge within the 5×10^5 function evaluation limit, resulting in terminal precision 3–4 orders of magnitude below the theoretical KKT convergence threshold. Case C's two-stage NLP pushes the variable count to ∼2409. Analytic Jacobian provision, adaptive mesh refinement strategies, and more efficient NLP solvers (e.g., IPOPT with exact second-order derivatives) represent potential pathways for overcoming this bottleneck.

Third, the introduction of path constraints fundamentally changes the coarse-search/NLP division of labor. In Cases A and B, the coarse search needed only to provide initial guesses within the NLP convergence basin. In Case C, trajectory safety must be independently verified through a two-layer enforcement strategy. The NLP-level constraint is applied at the 201 discrete collocation nodes with a conservative threshold $r \geq 0.28$ AU (0.03 AU above the physical limit to provide a margin for discretization error). The physical safety threshold $r_{\min} \geq 0.25$ AU, corresponding to a sail equilibrium temperature of approximately 464 K (191 °C)—well within the Kapton sustained-use limit of ∼620 K [45]—is enforced as a hard veto via high-fidelity ODE45 integration sampling at $\sim 10^4$ points per trajectory. Despite satisfying the NLP constraint, 40% of NLP-converged solutions in Case C fail this ODE45 safety check, with violated trajectories exhibiting perihelia as low as 0.0001 AU (equilibrium temperature exceeding 1300 K, far beyond the material limit). The discrepancy arises because the NLP constraint is evaluated only at the discrete collocation nodes, missing trajectory extrema that occur between nodes. This finding suggests that future implementations should embed the high-fidelity safety check within the optimization loop, creating a three-tier architecture (coarse search → NLP refinement → safety screening with feedback).

6.6. Dense Evaluation Budget Scenario

The comparative simulation results in Section 5 were obtained with algorithms at their standard per-iteration evaluation densities: FWA at approximately 540 evaluations per iteration (500 sparks + 40 Gaussian mutation) and PSO and DE at 100 per iteration ($N_p = 100$). To isolate the effects of the search mechanism from the evaluation budget, two complemen-

tary normalization experiments are conducted. This subsection presents the **Dense** scenario, in which PSO and DE population sizes are raised to 540 to match FWA's high evaluation density. All three algorithms thus operate at identical per-iteration evaluation counts, with three total budget levels—Low (20,000 evaluations, 37 iterations), Mid (60,000 evaluations, 111 iterations), and High (150,000 evaluations, 277 iterations). Ten independent runs (random seeds 43–52) are performed for each algorithm–budget combination.

Table 19 presents the coarse-search fitness J (mean $\pm$ standard deviation over 10 runs) and mean single-run computation time. Figure 11 visualizes the same benchmark as relative performance maps, making the cross-case ranking shifts easier to compare.

Table 19. Dense scenario ($N_p = 540$): coarse-search fitness J (mean $\pm$ std, 10 runs) and mean single-run computation time across three budget levels. Bold indicates lowest J per row.

Case	Budget	FWA		PSO		DE	
		J	CPU (s)	J	CPU (s)	J	CPU (s)
A	Low	**1347.1 ± 180.8**	581	1489.1 ± 189.7	428	2206.5 ± 221.5	1057
	Mid	**1257.5 ± 157.4**	1295	1286.1 ± 188.0	1020	1934.5 ± 129.8	2570
	High	**1301.0 ± 127.5**	3008	1353.1 ± 209.6	2427	1706.3 ± 109.1	4905
B	Low	555.9 ± 7.8	230	**551.0 ± 7.2**	172	643.9 ± 22.2	487
	Mid	558.8 ± 24.5	619	**543.0 ± 9.3**	455	625.0 ± 26.6	1473
	High	557.6 ± 9.8	1038	**537.4 ± 5.4**	1082	601.7 ± 21.1	3512
C P1	Low	0.15 ± 0.03	65	0.15 ± 0.03	60	0.38 ± 0.18	53
	Mid	0.16 ± 0.06	183	**0.14 ± 0.04**	182	0.26 ± 0.11	160
	High	0.15 ± 0.06	409	**0.14 ± 0.06**	401	0.24 ± 0.11	394
C P2	Low	17,631	80	**16,555**	90	21,941	64
	Mid	14,385	196	**12,222**	251	21,200	193
	High	11,882	503	**11,004**	566	19,790	496

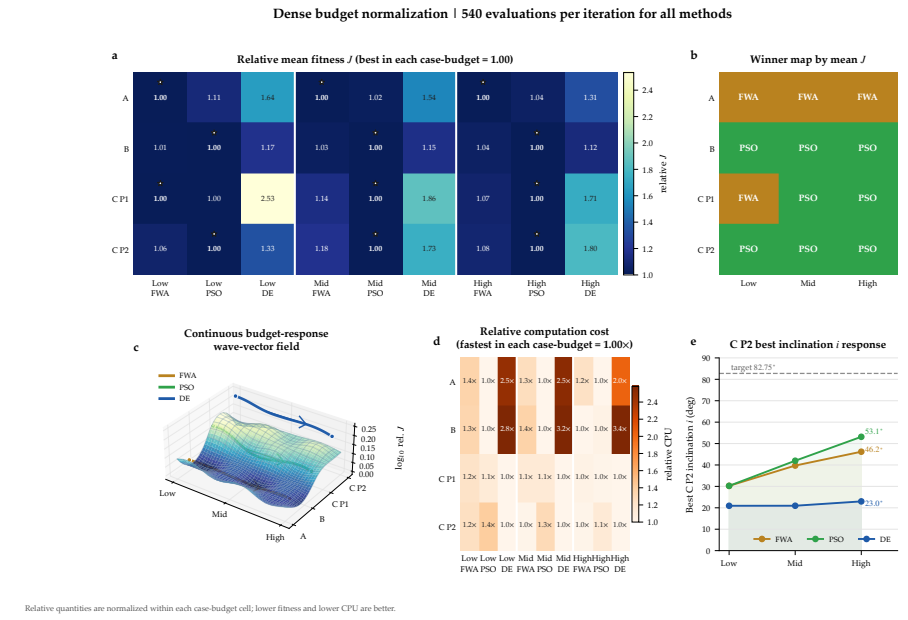


Figure 11. Dense budget-normalization summary (540 evaluations per iteration for all algorithms). (**a**) Relative mean fitness J, normalized so that the best algorithm in each case–budget cell equals 1.00 (marked with *); (**b**) winner map identifying the lowest-J algorithm per cell; (**c**) continuous budget-response curves showing $\log_{10}$ relative J versus budget level; (**d**) relative computation cost, normalized so that the fastest algorithm in each cell equals 1.00×; (**e**) best Case C Phase 2 terminal inclination versus budget, with the dashed line marking the 82.75° target. Color encodes algorithm: orange for FWA, green for PSO, blue for DE.

Several patterns emerge from the Dense benchmark. For Case A (Earth–Mars, ideal sail, single-modal), elevating PSO and DE to 540 evaluations per iteration does not close the performance gap with FWA. FWA achieves the lowest mean J across all three budget levels, with the Mid budget yielding the best overall $J_{\text{best}} = 1123.60$ (da/de/dw all within 2×10^{-3} of the target). The Mid-budget optimum—rather than the High—is consistent with the gradient-like quality distribution discussed in Section 6.1: FWA's explosion mechanism achieves sufficient exploration within 111 iterations; additional iterations (277) provide diminishing returns and may even slightly degrade mean fitness through over-exploration of local optima. PSO shows steady improvement with budget (1489 $\rightarrow$ 1286 $\rightarrow$ 1353), with the best terminal errors comparable to FWA at the High budget. DE, despite a 23% J reduction from Low to High (2207 $\rightarrow$ 1706), still exhibits terminal errors one to two orders of magnitude larger than FWA/PSO in semi-major axis and eccentricity.

For Case B (2016 HO_3, non-ideal sail, multi-modal), PSO achieves the lowest mean J across all budget levels under Dense normalization (High: 537.4 ± 5.4), with velocity errors converging to 4.6–4.8 km/s—the tightest range of any algorithm–case combination. A clear division of labor emerges when comparing FWA and PSO on terminal accuracy: FWA's position error collapses to sub-kilometer precision at Mid and High budgets (best runs: 6.1×10^1 km and <1 km, respectively), representing a four-order-of-magnitude improvement over DE. However, FWA's velocity error ($\sim$5.0 km/s) is consistently 0.2–0.4 km/s larger than PSO's best ($\sim$4.6–4.8 km/s). Because the Case B fitness function weights velocity matching more heavily than position matching, PSO's velocity advantage translates into a lower J despite FWA's superior position accuracy. This reveals a subtle interaction: fitness function design mediates which algorithmic strength is rewarded, and the observed algorithm ranking is partially an artifact of the relative weighting of position versus velocity objectives. DE continues to underperform markedly, with position errors exceeding 10^7 km—five orders of magnitude larger than FWA/PSO—reflecting the fundamental unsuitability of the DE/rand/1/bin perturbation strategy for this constrained, multi-modal landscape.

For Case C (SPO), the Phase 1 (orbital descent) results mirror Case A in that FWA and PSO achieve comparable precision (best J of 0.09–0.11 for both) with DE lagging significantly (best $J = 0.15$ at High). The FWA Mid-budget optimum recurs—$J_{\text{best}} = 0.09$ at 111 iterations versus 0.09 at 277 iterations—confirming that additional iterations do not improve extreme-value search quality for this 78-dimensional problem. Phase 2 (inclination raising) reveals the most instructive pattern. PSO High achieves the best overall Phase 2 result ($J_{\text{best}} = 4434$, $i = 53.15°$), outperforming FWA High ($J_{\text{best}} = 6732$, $i = 46.19°$) by a substantial margin. However, no algorithm approaches the target inclination of 83°, with the gap between the achieved result ($\sim$53°) and the target (83°) exceeding 29° for even the best run. This persistent shortfall, observed across all algorithms and budget levels, suggests that the difficulty is not algorithmic but structural: the two-phase B-spline encoding with 78 degrees of freedom may lack the representational capacity to encode the extreme cranking maneuver required for a $>50°$ inclination change under the $\beta = 0.1$ lightness number constraint. DE's Phase 2 results are qualitatively different from FWA/PSO: J values cluster narrowly around 17,900–19,200 with inclination limited to 21°–23°, reflecting the concentrated search distribution characteristic of this algorithm on complex landscapes.

The computation time data (Table 19) reveal algorithm-specific scaling patterns. PSO is the most computationally efficient for Cases A and B at Low and Mid budgets, with single-run times 26–38% shorter than FWA and 55–64% shorter than DE. At a High budget, however, FWA becomes the fastest for Case B (1038 s vs. 1082 s for PSO), and the three algorithms converge to similar per-run costs for Case C Phase 2 (496–566 s)—the additional

complexity of the two-phase problem equalizing their computational footprints. DE's computation time grows most rapidly with budget (e.g., Case A: 1057 → 4905 s, a 4.6× increase from Low to High), likely reflecting the higher cost of its pairwise difference-vector operations at large population sizes.

6.7. Fair Evaluation Budget Scenario

The Fair scenario takes the complementary normalization approach: FWA's effective population size is reduced to 100 to match the standard PSO/DE configuration. All three algorithms operate at identical per-iteration evaluation counts, with total budget levels being Low (20,000 evaluations, 200 iterations), Mid (60,000 evaluations, 600 iterations), and High (150,000 evaluations, 1500 iterations). Compared to the Dense scenario, the Fair configuration offers more iterations (200–1500 versus 37–277) but fewer evaluations per iteration, shifting the exploration–exploitation trade-off toward finer-grained iterative refinement at the expense of per-iteration sample density. All other experimental parameters (B-spline control points = 15, random seeds 43–52, 10 independent runs) remain identical to the Dense scenario.

Table 20 presents the coarse-search fitness J (mean ± std, 10 runs) and computation time. Figure 12 summarizes the corresponding relative rankings and highlights how the lower per-iteration evaluation density changes the algorithm ranking.

Table 20. Fair scenario ($N_p = 100$): coarse-search fitness J (mean ± std, 10 runs) and mean single-run computation time across three budget levels. Bold indicates lowest J per row.

Case	Budget	FWA		PSO		DE	
		J	CPU (s)	J	CPU (s)	J	CPU (s)
A	Low	2009.2 ± 567.0	259	**1405.1 ± 217.0**	367	2004.5 ± 85.6	787
	Mid	1521.6 ± 362.1	423	**1299.4 ± 227.5**	988	1662.7 ± 75.5	1750
	High	1608.2 ± 288.2	985	**1238.5 ± 173.8**	2393	1414.2 ± 64.4	3420
B	Low	597.7 ± 56.6	611	**550.5 ± 9.7**	161	650.0 ± 32.6	510
	Mid	591.0 ± 34.0	713	**540.5 ± 6.4**	465	623.7 ± 23.3	1494
	High	565.4 ± 30.5	732	**540.0 ± 8.1**	1139	598.7 ± 13.8	3772
C P1	Low	0.145 ± 0.033	362	**0.130 ± 0.030**	318	0.214 ± 0.056	289
	Mid	0.144 ± 0.034	621	**0.123 ± 0.024**	871	0.178 ± 0.017	837
	High	0.145 ± 0.033	517	**0.117 ± 0.020**	2140	0.145 ± 0.019	2204
C P2	Low	13,525 ± 3211	378	10,653 ± 2340	496	19,992 ± 969	358
	Mid	10,463 ± 3418	1232	11,061 ± 3869	1314	19,206 ± 908	1158
	High	11,077 ± 5334	2855	10,835 ± 4551	3493	18,656 ± 1092	3152

Reducing FWA to 100 evaluations per iteration substantially reshapes the performance landscape. For Case A, PSO now achieves the lowest mean J across all budget levels, with the High-budget mean (1238.5 ± 173.8) representing a 23% improvement over FWA's High-budget mean (1608.2 ± 288.2). The reversal is clear: under Dense normalization, FWA led by 4.0% at a High budget; under Fair normalization, PSO led by 23.0%. However, FWA's extreme-value search capability survives this budget compression: FWA's best single-run J (1126.76, High budget) actually improves upon its Dense-scenario counterpart (1139.46), and marginally surpasses PSO's best Fair result (1130.67). This behavior—high variance, high upside—is consistent across both normalization scenarios and confirms that FWA's explosion mechanism functions as a high-risk, high-reward strategy whose strength lies in locating exceptional solutions rather than achieving consistent average quality. PSO's social guidance, by contrast, produces a tighter J distribution (coefficient of variation 14.0% versus 17.9% for FWA at High budget), favoring reliability over upside potential.

For Case B, PSO dominates even more decisively under Fair normalization. Its mean J ranges from 550.5 (Low) to 540.0 (High), and its standard deviation (6.4–9.7) is the smallest of any algorithm–case–scenario combination, reflecting exceptional run-to-run consistency. Crucially, PSO's advantage in Case B is invariant to normalization scenario—it achieves the lowest mean J under both Dense and Fair configurations. This invariance excludes the possibility that PSO's Case B superiority in the original simulations (Section 5) was an artifact of budget asymmetry and instead confirms that the advantage is intrinsic to the social guidance mechanism's effectiveness for time-varying, multi-constraint rendezvous problems. FWA's position accuracy remains its signature strength even at reduced population density, but its velocity errors (∼5.0 km/s) consistently trail PSO's (∼4.6–4.8 km/s), and the fitness function's higher velocity weighting elevates PSO's J ranking.

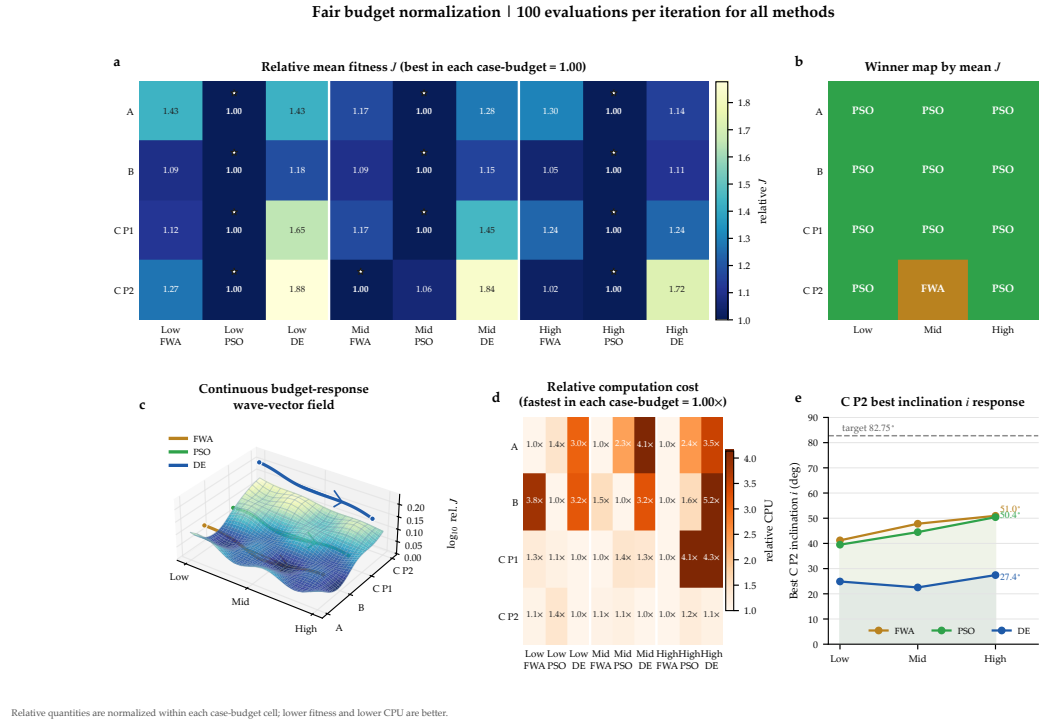


Figure 12. Fair budget-normalization summary (100 evaluations per iteration for all algorithms). (**a**) Relative mean fitness J, normalized so that the best algorithm in each case–budget cell equals 1.00 (marked with *); (**b**) winner map identifying the lowest-J algorithm per cell; (**c**) continuous budget-response curves showing $\log_{10}$ relative J versus budget level; (**d**) relative computation cost, normalized so that the fastest algorithm in each cell equals 1.00×; (**e**) Case C Phase 2 terminal inclination response under deeper but narrower iterative search, with the dashed line marking the 82.75° target. Color encodes algorithm: orange for FWA, green for PSO, blue for DE.

For Case C Phase 1, the Fair budget FWA retains superior extreme-value search: FWA's $J_{\text{best}} = 0.09$ is achieved at all three budget levels with an identical terminal semi-major axis ($a = 0.4800$ AU), whereas PSO requires the High budget to match this precision. However, PSO's mean J is consistently lower (e.g., 0.117 ± 0.020 versus 0.145 ± 0.033 at High), indicating that most PSO runs converge to near-optimal solutions while FWA exhibits a broader quality distribution with occasional exceptional runs. Phase 2 under Fair normalization produces the Case C highlight: FWA High achieves the global best Phase 2 J of 6395 ($i = 50.96°$, $t_f = 550.1$ d), outperforming both tPSO High (6573) and the best Dense-scenario Phase 2 result (PSO Dense High: 4434). The improved Phase 2 performance under Fair normalization—with more iterations (1500) but fewer per-iteration evaluations—suggests that for the severe multi-modality of inclination raising, iterative refinement depth matters more than per-iteration sample breadth. The additional 1223

iterations (Fair: 1500 versus Dense: 277) provide the search process more opportunities to escape local optima and progressively climb toward higher inclinations.

The computation time data under Fair normalization (Table 20) reveal a notable shift. PSO, the most efficient algorithm in the Dense scenario, becomes the second-slowest at a High budget across Cases A, B, and C. The reversal is largest for Case B High: PSO requires 1139 s while FWA requires 732 s—a 56% increase. This is because PSO's per-iteration cost is relatively insensitive to population size (each particle performs one evaluation regardless of N_p), so increasing iterations from 277 (Dense) to 1500 (Fair) multiplies PSO's total cost by approximately 5.4×, whereas FWA's cost—dominated by its explosion mechanism, which scales with population size—decreases when N_p is reduced from 540 to 100. The Fair scenario thus reverses the computational efficiency ranking: FWA becomes the fastest algorithm for Cases B and C at Mid and High budgets, while DE remains the slowest across nearly all configurations. An important caveat applies: the Case A FWA Fair runs were executed on 14-core hardware, while PSO/DE used six cores, so direct Case A CPU comparisons should factor in this hardware asymmetry.

6.8. Cross-Scenario Synthesis: Algorithm Adaptation Across Problem Types

The Dense and Fair normalization scenarios, together with the original hybrid simulations (Section 5), constitute a three-point evaluation of budget sensitivity. Table 21 distills the cross-scenario rankings.

Table 21. Algorithm ranking (J mean, Low–High budget range) across the three evaluation scenarios: original (FWA 540, PSO/DE 100 per iteration), Dense (all 540), and Fair (all 100). Rankings derived from the mean J values reported in Sections 5, 6.6 and 6.7.

Case	Scenario	1st (Lowest J)	2nd	3rd
A	Original	PSO	FWA	DE
	Dense	FWA	PSO	DE
	Fair	PSO	DE	FWA
B	Original	PSO	FWA	DE
	Dense	PSO	FWA	DE
	Fair	PSO	FWA	DE
C P1	Original	FWA/PSO	–	DE
	Dense	PSO	FWA	DE
	Fair	PSO	FWA	DE
C P2	Dense	PSO	FWA	DE
	Fair	FWA/PSO	–	DE

Case A original ranking by coarse J mean: PSO (1328.2) ≈ FWA (1345.8) ≪ DE (1670.2). NLP convergence rates are statistically indistinguishable (52%, 50%, 53%).

Three cross-scenario regularities emerge. First, PSO's dominance for Case B is scenario-invariant. Under all three configurations (original, Dense, Fair), PSO achieves the lowest mean J with the smallest standard deviation. This invariant ranking constitutes the strongest evidence in the present study that a specific algorithm–problem pairing reflects a genuine mechanistic match rather than an experimental confound. The social guidance mechanism of PSO—where each particle's velocity update incorporates both personal and global best positions—appears intrinsically well-suited to the six-dimensional, time-varying constraint topology of the 2016 HO_3 rendezvous problem, where the target's moving ephemeris demands continuous trajectory adjustments that benefit from the swarm's collective information sharing.

Second, FWA's performance is strongly budget-density-dependent. FWA's ranking moves from first (Dense Case A) to third (Fair Case A), a swing of two positions driven entirely by per-iteration evaluation density. This sensitivity confirms that

FWA's explosion mechanism—which generates 500 sparks per iteration in its standard configuration—derives a substantial fraction of its search power from the volume of parallel evaluations rather than from any qualitative superiority of the explosion sampling strategy over swarm or differential-evolution sampling. However, FWA's extreme-value search capability—its ability to find the single best solution across all runs—is robust to budget compression. In both Fair Case A (best $J = 1126.76$) and Fair Case C Phase 2 (best $J = 6395$), FWA's best single-run result equals or exceeds the best result from any other algorithm–scenario combination. This pattern suggests that FWA is optimally deployed in mission design contexts where discovering the single best trajectory—rather than achieving reliable average performance—is the overriding objective.

Third, DE's ranking is systematically lowest across all scenarios and cases. The sole exception occurs in Fair Case C Phase 2, where DE's mean J (18,656–19,992) is misleadingly competitive because the adaptation of the fitness function to the Phase 2 problem structure—penalizing failure to raise inclination—produces a compressed scoring range where all algorithms score poorly. DE's consistent bottom ranking should not be interpreted as evidence that differential evolution is categorically unsuitable for trajectory optimization. Rather, it reflects the specific DE configuration tested (DE/rand/1/bin, $F = 0.8$, $CR = 0.9$) and, more fundamentally, the mismatch between the rand/1 mutation strategy's localized perturbation scale and the global exploration demands of interplanetary trajectory problems. The concentrated coarse J distribution produced by DE (coefficient of variation 6.4% for Case A versus 14.9%–15.9% for FWA/PSO) indicates that its search concentrates in a narrow region, which is advantageous when the NLP convergence basin is large (enabling high convergence rates from concentrated samples) but disadvantageous when the basin is small or the landscape is multi-modal (causing the concentrated sample to miss the basin entirely). For low-dimensional problems with known, benign landscapes—or for rapid prototyping where computational speed is prioritized over solution optimality—DE's parameter parsimony (only two tuning parameters) and predictable convergence behavior remain attractive properties. Adaptive variants such as JADE and SHADE, which self-tune F and CR during the search, may behave differently and warrant separate evaluation.

Synthesizing the evidence from the three case studies, the cold-start comparison, and the budget-normalization analysis, a consistent picture emerges: in practice, no single metaheuristic dominates across all problem types; rather, each algorithm's suitability is contingent on the constraint structure and modality of the specific optimization landscape. This empirical guideline is consistent with the No Free Lunch theorems of Wolpert and Macready [47], which establish that, averaged over all possible problems, no optimizer outperforms any other; our claim is therefore problem-class-specific rather than a theoretical universal. Three practically relevant guidelines can be distilled:

For simple, single-modal problems with large convergence basins (Case A), DE's parameter-parsimonious nature and fast per-iteration cost make it a computationally efficient default for rapid prototyping, although FWA or PSO should be preferred when terminal precision requirements are stringent (da/de/dw within 10^{-3} of target). In such problems, the choice of algorithm is less critical than the quality of the NLP refinement phase.

For multi-modal problems with diverse, qualitatively distinct solution families (Case B), FWA's explosion mechanism provides a distinct advantage in discovering superior solution families that swarm-based or differential-evolution approaches systematically miss, making it the preferred choice when the primary objective is to minimize flight time or fuel consumption. PSO's social guidance mechanism offers superior run-to-run consistency, which is valuable when predictable, repeatable performance is required for mission planning, but at the cost of potentially missing the global optimum.

For Case C safety-constrained transfers, in this severely multi-modal setting, algorithm selection involves a trade-off between exploration breadth and constraint compliance. PSO's social conservatism yields 100% safety compliance, while FWA's aggressive exploration achieves superior coarse-search exploration but at a 40% safety violation rate. The optimal choice depends on the mission's risk tolerance: for operational missions where constraint violation is unacceptable, PSO is the safer choice; for conceptual design studies exploring the performance envelope, FWA's broader exploration may reveal design possibilities that are invisible to more conservative search strategies.

In summary, the hybrid optimization framework itself is validated as a general-purpose strategy applicable across all three problem types, regardless of which specific metaheuristic is employed. The metaheuristic acts as a modular front-end that can be selected and configured to match the problem's constraint structure and the designer's priorities (exploration versus reliability, extreme-value search versus consistent quality). The principal engineering contribution of the present work is therefore a modular framework with FWA-assisted multi-modal discovery, in which the metaheuristic front-end can be selected and configured to match the problem's constraint structure—rather than the superiority of any single algorithm. Future research should extend the framework to incorporate machine learning techniques for real-time initial guess generation, adaptive mesh refinement strategies for high-dimensional NLP problems, and embedded safety-verification loops for path-constrained mission scenarios.

7. Conclusions

This paper introduced a hybrid optimization strategy for solar sail trajectory design in which population-based coarse search is coupled with direct-collocation NLP refinement. The results across the benchmark missions show that the two layers have different roles: the metaheuristic stage explores feasible dynamic structures, while the NLP stage converts a promising structure into a locally consistent trajectory satisfying the terminal and path constraints.

The cold-start comparison further shows that warm starting is necessary for difficult solar sail transfers. The purpose of the coarse search is therefore not simply to find the point closest to the desired terminal state. A more useful coarse solution is one that places the NLP problem in a basin where the dynamics, constraints, and boundary conditions can be refined effectively. This changes the role of metaheuristics from the search for a final solution to NLP-enabling initialization. The dynamical model adopted here is restricted to the two-body problem perturbed by solar radiation pressure; planetary gravitation (including the Earth's sphere of influence), occultation/eclipses, and solar J2 are neglected. This level of fidelity suffices to demonstrate the hybrid framework's convergence behavior and the inter-algorithm comparison, but future work should extend the strategy to higher-fidelity ephemeris models that incorporate these effects and to mission-specific constraint sets.

Author Contributions: Conceptualization and methodology, G.H.; writing—original draft preparation and simulation, Y.T.; software, data curation, and writing—review and editing, H.Y.; validation and formal analysis, J.W.; supervision and project administration, Z.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (No. 12502409, 12502411), State Key Laboratory of Advanced Space Propulsion (No. 2025SP0106) and Innovation Research Foundation of NUDT (25-ZZCX-JDZ-44, ZK25-70).

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Acknowledgments: During the preparation of this manuscript, the authors used GLM-5 for the purposes of grammar and readability improvement. The authors have reviewed and edited the output and take full responsibility for the content of this publication.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Macdonald, M.; McInnes, C. Solar Sail Science Mission Applications and Advancement. *Adv. Space Res.* **2011**, *48*, 1702–1716. [CrossRef]
2. Fu, B.; Sperber, E.; Eke, F. Solar Sail Technology—A State of the Art Review. *Prog. Aerosp. Sci.* **2016**, *86*, 1–19. [CrossRef]
3. Bassetto, M.; Niccolai, L.; Quarta, A.A.; Mengali, G. A Comprehensive Review of Electric Solar Wind Sail Concept and Its Applications. *Prog. Aerosp. Sci.* **2022**, *128*, 100768. [CrossRef]
4. Mori, O.; Sawada, H.; Funase, R.; Morimoto, M.; Endo, T.; Yamamoto, T.; Tsuda, Y.; Kawakatsu, Y.; Kawaguchi, J.; Miyazaki, Y.; et al. First Solar Power Sail Demonstration by IKAROS. *Trans. Jpn. Soc. Aeronaut. Space Sci. Aerosp. Technol. Jpn.* **2010**, *8*, To_4_25–To_4_31. [CrossRef] [PubMed]
5. Tsuda, Y.; Mori, O.; Funase, R.; Sawada, H.; Yamamoto, T.; Saiki, T.; Endo, T.; Yonekura, K.; Hoshino, H.; Kawaguchi, J. Achievement of IKAROS—Japanese Deep Space Solar Sail Demonstration Mission. *Acta Astronaut.* **2013**, *82*, 183–188. [CrossRef]
6. Tsuda, Y.; Mori, O.; Funase, R.; Sawada, H.; Yamamoto, T.; Saiki, T.; Endo, T.; Kawaguchi, J. Flight Status of IKAROS Deep Space Solar Sail Demonstrator. *Acta Astronaut.* **2011**, *69*, 833–840. [CrossRef]
7. Spencer, D.A.; Betts, B.; Bellardo, J.M.; Diaz, A.; Plante, B.; Mansell, J.R. The LightSail 2 Solar Sailing Technology Demonstration. *Adv. Space Res.* **2021**, *67*, 2878–2889. [CrossRef]
8. Mansell, J.; Spencer, D.A.; Plante, B.; Diaz, A.; Fernandez, M.; Bellardo, J.; Betts, B.; Nye, B. Orbit and Attitude Performance of the LightSail 2 Solar Sail Spacecraft. In Proceedings of the AIAA Scitech 2020 Forum, Orlando, FL, USA, 6–10 January 2020. [CrossRef]
9. Heiligers, J.; Hiddink, S.; Noomen, R.; McInnes, C.R. Solar Sail Lyapunov and Halo Orbits in the Earth–Moon Three-Body Problem. *Acta Astronaut.* **2015**, *116*, 25–35. [CrossRef]
10. Simo, J.; McInnes, C.R. Solar Sail Orbits at the Earth–Moon Libration Points. *Commun. Nonlinear Sci. Numer. Simul.* **2009**, *14*, 4191–4196. [CrossRef]
11. Huo, M.; Yu, Z.; Liu, H.; Zhao, C.; Lin, T.; Song, Z.; Qi, N. Initial Three-Dimensional Trajectory Design for Solar Sails Using Bezier Shaping Approach. *IEEE Access* **2019**, *7*, 150842–150850. [CrossRef]
12. Peloni, A.; Ceriotti, M.; Dachwald, B. Solar-Sail Trajectory Design for a Multiple Near-Earth-Asteroid Rendezvous Mission. *J. Guid. Control Dyn.* **2016**, *39*, 2712–2724. [CrossRef]
13. Orphee, J.; Diedrich, B.; Stiltner, B.; Becker, C.; Heaton, A. Solar Sail Attitude Control System for the NASA Near Earth Asteroid Scout Mission. In Proceedings of the Fourth International Symposium on Solar Sailing, Kyoto, Japan, 17–20 January 2017.
14. Gong, S.; Li, J. Equilibria near Asteroids for Solar Sails with Reflection Control Devices. *Astrophys. Space Sci.* **2015**, *355*, 213–223. [CrossRef]
15. Peloni, A.; Dachwald, B.; Ceriotti, M. Multiple Near-Earth Asteroid Rendezvous Mission: Solar-sailing Options. *Adv. Space Res.* **2018**, *62*, 2084–2098. [CrossRef]
16. Zamora-Caballero, G.; Vazquez, R.; Topputo, F. Solar Sail Trajectory Optimization for Asteroid Flyby and Rendezvous. *J. Guid. Control Dyn.* **2025**, *48*, 2257–2271. [CrossRef]
17. Wie, B.; Murphy, D. Solar-Sail Attitude Control Design for a Flight Validation Mission. *J. Spacecr. Rockets* **2007**, *44*, 809–821. [CrossRef]
18. Fitzgerald, R.M. Characterizing Minimum-Time Solar Sail Geostationary Orbit Transfers Using Pseudospectral Optimal Control. *J. Spacecr. Rockets* **2021**, *58*, 997–1009. [CrossRef]
19. Ross, I.M.; Fahroo, F. Legendre Pseudospectral Approximations of Optimal Control Problems. In *New Trends in Nonlinear Dynamics and Control and Their Applications*; Kang, W., Borges, C., Xiao, M., Eds.; Springer: Berlin/Heidelberg, Germany, 2004; Volume 295, pp. 327–342. [CrossRef]
20. Fahroo, F.; Ross, I.M. Direct Trajectory Optimization by a Chebyshev Pseudospectral Method. *J. Guid. Control Dyn.* **2002**, *25*, 160–166. [CrossRef] [PubMed]
21. Herman, A.L.; Conway, B.A. Direct Optimization Using Collocation Based on High-Order Gauss-Lobatto Quadrature Rules. *J. Guid. Control Dyn.* **1996**, *19*, 592–599. [CrossRef] [PubMed]
22. Hargraves, C.; Paris, S. Direct Trajectory Optimization Using Nonlinear Programming and Collocation. *J. Guid. Control Dyn.* **1987**, *10*, 338–342. [CrossRef] [PubMed]

23. Probst, A.; Anderson, T.; Farrish, A.; Kjellstrand, C.; Newheart, A.; Thaller, S.; Young, S.; Rankin, K.; Akhavan-Tafti, M.; Chartier, A.; et al. Sun Sailing Polar Orbiting Telescope (SunSPOT): A Solar Polar Imaging Mission Design. *Adv. Space Res.* **2022**, *70*, 510–522. [CrossRef]
24. Yao, Z.; Chu, Y.; Shi, P.; Li, W.; Huang, R.; Yuan, C.; Gong, S. Orbit Design and Attitude Control of Diffractive Sail and Validation with Solar Polar Mission. *Acta Astronaut.* **2024**, *218*, 195–208. [CrossRef]
25. Melton, R.G. Hybrid Methods for Determining Time-Optimal, Constrained Spacecraft Reorientation Maneuvers. *Acta Astronaut.* **2014**, *94*, 294–301. [CrossRef]
26. He, G.; Conte, D.; Melton, R.G.; Spencer, D.B. Fireworks Algorithm Applied to Trajectory Design for Earth to Lunar Halo Orbits. *J. Spacecr. Rockets* **2020**, *57*, 235–246. [CrossRef]
27. Pontani, M.; Conway, B.A. Particle Swarm Optimization Applied to Impulsive Orbital Transfers. *Acta Astronaut.* **2012**, *74*, 141–155. [CrossRef]
28. Luo, Y.-Z.; Zhou, L.-N. Asteroid Rendezvous Mission Design Using Multiobjective Particle Swarm Optimization. *Math. Probl. Eng.* **2014**, *2014*, 1–13. [CrossRef]
29. Wang, M.; Song, Z.; Dai, G.; Peng, L.; Zheng, C. Asteroids Exploration Trajectory Optimal Design with Differential Evolution Based on Mixed Coding. *Int. J. Distrib. Sens. Netw.* **2015**, *11*, 827987. [CrossRef]
30. Heiligers, J.; Fernandez, J.M.; Stohlman, O.R.; Wilkie, W.K. Trajectory Design for a Solar-Sail Mission to Asteroid 2016 HO_3. *Astrodynamics* **2019**, *3*, 231–246. [CrossRef]
31. Macdonald, M.; Hughes, G.W.; McInnes, C.R.; Lyngvi, A.; Falkner, P.; Atzei, A. Solar Polar Orbiter: A Solar Sail Technology Reference Study. *J. Spacecr. Rockets* **2006**, *43*, 960–972. [CrossRef]
32. Luo, X.; Liu, C.; Yue, X. Improved-PPO-Based Adaptive Safety-Critical Control for Orbit-Attitude Takeover of Failed Spacecraft With Cellular Satellites. *IEEE Trans. Autom. Sci. Eng.* **2026**, *23*, 10872–10882. [CrossRef]
33. Ceriotti, M.; McInnes, C.R. Generation of Optimal Trajectories for Earth Hybrid Pole Sitters. *J. Guid. Control Dyn.* **2011**, *34*, 847–859. [CrossRef]
34. McInnes, C.R. *Solar Sailing: Technology, Dynamics and Mission Applications*; Springer: Berlin/Heidelberg, Germany, 2004. [CrossRef]
35. Heaton, A.; Miller, K.C.; Ahmad, N. Near Earth Asteroid Scout Solar Sail Thrust and Torque Model. In Proceedings of the Fourth International Symposium on Solar Sailing (ISSS), Kyoto, Japan, 17–20 January 2017.
36. Li, J.; Tan, Y. A Comprehensive Review of the Fireworks Algorithm. *ACM Comput. Surv.* **2020**, *52*, 1–28. [CrossRef]
37. Kennedy, J.; Eberhart, R. Particle Swarm Optimization. In Proceedings of the IEEE International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948. [CrossRef]
38. Storn, R.; Price, K. Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *J. Glob. Optim.* **1997**, *11*, 341–359. [CrossRef]
39. Tan, Y.; Zhu, Y. Fireworks Algorithm for Optimization. In *Advances in Swarm Intelligence, Proceedings of the First International Conference on Swarm Intelligence (ICSI), Beijing, China, 12–15 June 2010*; Tan, Y., Shi, Y., Tan, K.C., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2010; Volume 6145, pp. 355–364. [CrossRef]
40. Shi, Y.; Eberhart, R. A Modified Particle Swarm Optimizer. In Proceedings of the 1998 IEEE International Conference on Evolutionary Computation, Anchorage, AK, USA, 4–9 May 1998; pp. 69–73. [CrossRef]
41. McInnes, C.R. Solar Sailing: Orbital Mechanics and Mission Applications. *Adv. Space Res.* **2003**, *31*, 1971–1980. [CrossRef]
42. Dachwald, B.; Ohndorf, A.; Wie, B. Solar Sail Trajectory Optimization for the Solar Polar Imager (SPI) Mission. In Proceedings of the AIAA/AAS Astrodynamics Specialist Conference, Keystone, CO, USA, 21–24 August 2006. [CrossRef]
43. Kezerashvili, R.Y. Solar Sail: Materials and Space Environmental Effects. In *Advances in Solar Sailing*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 573–592. [CrossRef]
44. Kang, J.H.; Gordon, K.; Bryant, R.; Stohlman, O.; Wilkie, W.K.; Stark, A.; Barfield, R.; Sindle, B.; Finckenor, M.; Craven, P. *Durability Characterization of Mechanical Interfaces in Solar Sail Membrane Structures*; Technical Report 20200002699; NASA: Washington, DC, USA, 2020.
45. DuPont. *Kapton® HN Polyimide Film—Technical Data Sheet*; Bulletin H-38492-2; DuPont: Wilmington, DE, USA, 1996.
46. Caruso, A.; Quarta, A.A.; Mengali, G.; Ceriotti, M. Shape-Based Approach for Solar Sail Trajectory Optimization. *Aerosp. Sci. Technol.* **2020**, *107*, 106363. [CrossRef]
47. Wolpert, D.H.; Macready, W.G. No Free Lunch Theorems for Optimization. *IEEE Trans. Evol. Comput.* **1997**, *1*, 67–82. [CrossRef]

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.