Computer Methods in Applied Mechanics and Engineering 446 (2025) 118247

Contents lists available at ScienceDirect

Comput. Methods Appl. Mech. Engrg.

journal homepage: www.elsevier.com/locate/cma

Multi-objective mission planning for solar sails and swarm networks

Mostafa Sedky [a,1], Maya Horii [a,1], Thomas Hosmer [a], Kristofer Pister [b], Tarek Zohdi [a,*]

[a] *Department of Mechanical Engineering, University of California, Berkeley, CA, 94720, USA*
[b] *Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA, 94720, USA*

ARTICLE INFO

Keywords:
Solar sails
Dynamics
Trajectory-optimization
Evolutionary-optimization

ABSTRACT

Solar sails, which use optical pressure from the sun as propulsion, have great potential to be an inexpensive and sustainable way of exploring the universe. Recent advancements in the semiconductor industry have given rise to small and light actuators, sensors, and cameras, which make it possible to design high-performance, low-cost solar sails. For example, the Berkeley Low-cost Interplanetary Solar Sail (BLISS) is a sail design that takes advantage of lightweight electronics to enable a high sail area-to-mass ratio. Being low-cost and propellant-free, this design is well suited for rapid swarm exploration of near-Earth objects and long-duration missions. Since many solar sail missions are launched as secondary loads to a primary mission, rapid identification of feasible trajectories for changing schedules and mission parameters is needed. This paper develops a navigation strategy framework to address this need, designed to be lightweight and fast enough to run on a laptop. It consists of a general and customizable model for solar sail flight and a corresponding optimization framework integrating genetic algorithms with a coordinate descent method. This method is able to minimize flight time, control hover duration, manage closest sail approach to the Sun, and match approach velocities by specifying sail controls over time. Using BLISS as our solar sail design, we demonstrate this framework in three cases: sail-asteroid rendezvous, swarm network communication, and solar system escape. The goal is to allow researchers to utilize this framework to efficiently explore the design space to identify navigation strategies for their sail designs and missions.

1. Introduction

1.1. NEO Exploration

Deep space exploration has the potential for significant scientific returns and technological advancement. Within this field, near-Earth objects (NEOs) are a particular area of interest: defined as asteroids with a perihelion distance less than 1.3 AU, their study may reveal crucial information about the early formation of the solar system and the origins of life [1,2]. Cataloging and understanding NEOs is also crucial for planetary defense, as collisions with Earth could lead to significant destruction and loss of life [1]. NEOs could

[*] Corresponding author.
E-mail address: zohdi@berkeley.edu (T. Zohdi).
[1] These authors contributed equally to this work.

https://doi.org/10.1016/j.cma.2025.118247
Received 19 May 2025; Received in revised form 1 July 2025; Accepted 15 July 2025

also be mined for resources like metal and water (for return to Earth or for in-space usage) [3], and could serve as testing grounds for future Mars missions [4].

Despite the benefits of NEO study, the majority of NEO investigation has been limited to distant, ground-based observations [1]. Asteroid detection is done largely by analyzing sequences of images taken in the optical and IR wavelengths to find the reflected light, which may be faint and spread across multiple pixels due to their rapid movement. This process can be augmented with advanced detection algorithms, multiple observation sources, or time-delayed multi-wavelength imaging [2]. Once identified, spectral analysis from ground observations can indicate the likely composition of NEOs, and their sizes can be estimated by observing brightness over time or using delay-doppler radar. However, these methods have limited certainty and applicability. Delay-doppler radar, for example, is only able to characterize close, slow objects [2]. Additionally, size estimates are more commonly made via brightness measurements, which require estimating albedo (an attribute describing the reflectivity of the NEO) and are therefore often uncertain [2,5]. Other challenges in NEO study and discovery include limited observation windows of Earth-based instruments, which generally can only be used when facing away from the Sun. NEOs with long orbit periods are also less likely to have been yet observed, and optical measurements are biased towards detecting higher albedo objects [5,6]. Grav et al. estimated that the population of NEOs greater than 140 m in diameter is $\sim 38\,\%$ discovered as of 2022 [7]. In comparison, a US Congressional mandate (The George E. Brown, Jr. Near-Earth Object Survey Act) aimed to have $> 90\,\%$ of NEOs cataloged by 2020.

For more reliable and detailed information, spacecraft exploration is required. A relatively small number of missions have made nearby observations of asteroids. This includes Galileo, the first mission to capture close-by (∼1600 km) images of an asteroid in 1991, and more recent missions like OSIRIS-REX, which returned a sample of the NEO Bennu (101955) to Earth for study in September of 2023 [8,9]. These missions have been able to uncover information previously unobservable from Earth, including the surprise discovery of the first confirmed asteroid satellite by the mission Galileo, unexpected surface characteristics on asteroid 433 Eros captured by the NEAR-Shoemaker spacecraft, and compelling evidence for the asteroid 4 Vesta being the origin of a family of meteorites found on Earth from the mission Dawn [8,10].

1.2. Solar sails

Solar sails, first conceptualized as early as the 1920s [11], use optical pressure from the Sun as propulsion. In comparison to traditional propellant-based systems, they have the capability for continuous thrust, unlimited specific impulse, and potentially reduced launch costs. This makes them particularly suited to long-duration and high-energy missions which would otherwise be difficult and costly. For NEO exploration, where there is a large catalog of objects to survey, this low cost is particularly important.

As the optical pressure of the Sun is relatively small, a solar sail's ability to maneuver relies on having a large sail relative to the craft's mass. This property is quantified by the lightness number, β, which is a ratio of the maximum possible acceleration from solar radiation to the Sun's gravity. The lightness number is defined as

$$\beta = \frac{2A_o P_E R_E^2}{G m_{Sun} m_s}. \tag{1.1}$$

where P_E is the solar radiation pressure at 1 Astronomical Unit (AU), A_o is area of the sail, G is the gravitational constant, m_{Sun} is the mass of the Sun, m_s is the mass of the solar sail, and R_E is the radius of Earth's orbit, approximately equal to 1 AU. Since gravitational force and optical pressure are both proportional to the inverse squared distance from the Sun, the lightness number is a property of the sail, independent of the Sun-sail distance, and is a critical metric of solar sail performance. As seen in Eq. (1.1), β is also proportional to the area-to-mass ratio of the craft. The highest lightness numbers of any solar sail missions so far have been around 0.01 [12].

To achieve the lightness numbers required for sufficient control, many solar sail designs with heavy payloads require large sails on the order of 100-400 m^2. Early development tended toward larger sail sizes: Znamya-2 launched in 1993 and deployed a proof-of-concept sail with a 20 m diameter, the ESA and DLR carried out ground tests with a 330.5 m^2 sail in 1999, ATK and L'Garde also carried out ground tests with a 400 m^2 sail in 2004, and the first successful interplanetary solar sail mission in 2010 (IKAROS) had a sail with an approximate area of 196 m^2 [13,14]. For reference, IKAROS had an initial mass of 307 kg [13].

More recently, many have begun to design smaller solar sail crafts which are able to use smaller sail areas. For example, Lightsail-2 was a 3U CubeSat with a mass of 4.93kg and a 32m^2 sail, and successfully demonstrated control using solar pressure during its 2019–2022 mission [15]. The small size of Lightsail-2 enabled it to have a lightness number of 0.0098, one of the highest of solar sail missions to date [12].

1.3. BLISS

Following the trend toward decreasing solar sail craft size to improve cost and maneuverability, Alvara et. al. developed the Berkeley Low-cost Interplanetary Solar Sail (BLISS) [16]. BLISS was developed with miniaturized technology to achieve a sail load of 10 g/m^2 on a sail area of just over 1 m^2, corresponding to a lightness number of $\beta = 0.16$ (an order of magnitude greater than Lightsail-2) [12]. The cost per BLISS craft is less than $1000 USD, making it unmatched in terms of payload and price.

BLISS currently relies on optical communication through a semiconductor laser transmitter [16], which requires being close to a target to send and receive data. If regular communication with Earth was required, the sail's range would be limited. However, for missions that do not require regular communication, BLISS could simply return to Earth to transmit data after collection. Due to its low cost and weight, BLISS is also well suited to be deployed as a swarm of sails. Then, network communication within the swarm can enable enhanced communication time without increasing the payload of the individual BLISS crafts.

1.4. Trajectory optimization

Though much work has proposed trajectory design methods for solar sails [17–20], only one of the launched solar sail missions' primary aim was a target fly-by: NASA's NEA Scout, which launched in 2022 and intended to photograph and study Asteroid 2020 GE, but lost communication after launch (other missions were meant primarily for sail deployment demonstrations and/or sail performance and solar radiation pressure measurements) [21]. Like many solar sail missions, NEA Scout was a secondary load and was therefore constrained by the primary mission's initial launch trajectory and schedule changes. NEA Scout targeted a path that would eventually use a lunar gravity assist to reach the asteroid. To identify a trajectory from the initial launch to that lunar gravity assist, the NEA Scout team created a large database of paths starting from possible initial trajectories (forward) and paths ending at the desired lunar gravity assists (backward) using a variety of control laws and a simplified model to check for feasibility. Matching forward-backward paths would then be used to seed the initial condition of a more sophisticated model. This necessitated collecting on the order of a million paths for both the forward and backward paths, highlighting the need for fast models and efficient sampling in navigation planning. A similar approach is taken in Vergaaij et. al [22], where grid search and genetic algorithm approaches are used to generate forward and backward trajectories, with good matches used as an initial guess for an optimal control solver (here, a pseudospectral collocation method).

In this paper, we use a genetic algorithm (GA) to sample the design space widely and identify areas of low-cost, then use a coordinate descent algorithm to hone in on a more precise trajectory. GAs are a non-derivative optimization approach, suitable for exploring the design space of complex problems with non-convex objective functions [23–28]. This framework is highly generalizable — additional user-set goals or constraints can easily be added without requiring additional changes to the GA or coordinate descent algorithm.

1.5. Objectives

With the general advantages of solar sails (propellant-free, continuous thrust, infinite specific impulse) being well-suited to long-duration, high-energy missions, and the specific advantages of BLISS (maneuverability, extremely low cost) being additionally well-suited for rapid exploration, higher-risk exploration, and swarm operations, we design three mission objectives tailored to these strengths. In the first, a single-sail NEO rendezvous and return to Earth takes advantage of the high-energy capabilities of solar sails in this multi-objective mission. In the second, a sail swarm is used for space exploration in a solar system escape mission, which takes advantage of the long-duration as well as the swarm capabilities of BLISS (communication at this distance would be a challenge — while it is beyond the scope of this work to design a full solution, there are existing networks for interstellar communication, like the operational Deep Space Network [29,30]). In the third, a sail swarm uses network communication to explore NEOs and return data to Earth using BLISS' suitability to swarm operations and rapid exploration.

We develop a model of solar sail flight dynamics and swarm communication, and we optimize the navigation strategies for each mission objective with a genetic and coordinate descent algorithm. Both the model and optimization method are highly generalizable and can be modified for different solar sail builds and mission objectives.

The framework we describe in this work is designed to run quickly on a laptop, allowing for efficient and cheap prototyping for navigation strategies before fine tuning a more complex model. The goal is to allow researchers to utilize this framework to efficiently explore the design space for navigation strategies of interest for their designed sails.

2. Methods

In this section, we discuss the formulation of our solar sail model, the proposed trajectory design space, and the optimization of control for path planning. Our model (described in detail in the section Solar Sail Model) uses a forward-Euler time stepping method to solve for the position and velocity of the solar sail, taking into account gravitational forces and the optical pressure force (as a function of sail position). In the section Trajectory Design Space, we describe the design space for sail control in terms of the design variables and limits. Lastly, we define our cost functions and perform optimization with a combined genetic algorithm and coordinate descent approach as described in the section Optimization Approach.

We will demonstrate the results of this approach for three scenarios:

- Case 1: A round-trip mission for a single solar sail to rendezvous with a NEO and return to Earth.
- Case 2: A solar system exit for a swarm of solar sails for interstellar space exploration.
- Case 3: A NEO rendezvous and data return mission using a network of solar sails.

The code for this work can be found in https://github.com/Mssedky/MultiObjective-Solar-Sails.git, and the run-times on a single-thread Python implementation on a standard laptop computer are typically on the order of several minutes for Cases 1 and 3, and up to several hours for Case 2 depending on the number of sails. An alternative single-thread C + + implementation is also provided for Case 1 to allow for faster run times (also on the order of several minutes) for smaller time steps and tighter rendezvous tolerances which typically require more iterations.

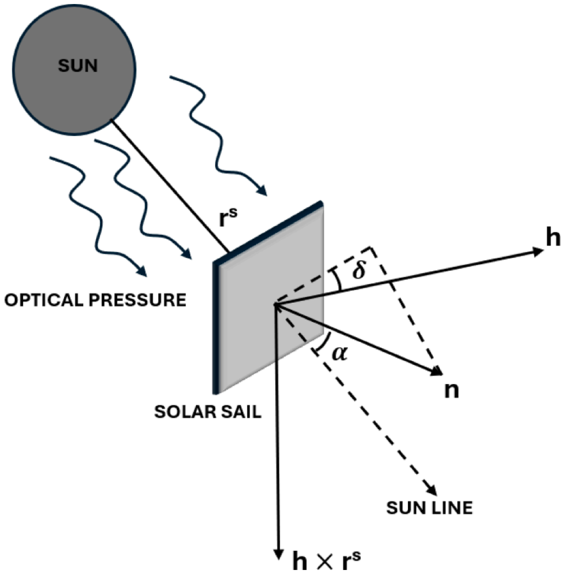

Fig. 1. Diagram of the system.

2.1. Solar sail model

2.1.1. Solar sail position

We define the solar sail position as seen in Fig. 1 using a universal, fixed coordinate system centered at the Sun, where:

- $\boldsymbol{r}^s$ is the position vector of the solar sail relative to the Sun.
- $\boldsymbol{u}^s = \frac{\boldsymbol{r}^s}{||\boldsymbol{r}^s||}$ is the unit vector pointing from the Sun to the solar sail.
- α is the cone angle, representing the angle between the sail area normal and the sun line.
- δ is the clock angle, representing the angle between the projection of the sail area normal onto a plane normal to the sun line and an in-plane reference direction (here, **h**).
- $\boldsymbol{h}$ is a unit vector in the direction of the orbital angular momentum, defined as $\boldsymbol{h} = \frac{\boldsymbol{r}^s \times \boldsymbol{v}^s}{||\boldsymbol{r}^s \times \boldsymbol{v}^s||}$, where v^s is the solar sail velocity.
- $\boldsymbol{n}$ is the outward normal from the solar sail (away from the Sun).

We define $\boldsymbol{n}$ in terms of the angles α and δ in Fig. 1, where $\boldsymbol{q} = \boldsymbol{h} \times \boldsymbol{u}^s$ [31]. $\boldsymbol{n}$ is a unit vector regardless of clock and cone angle values.

$$\boldsymbol{n} = \cos(\alpha)\,\boldsymbol{u}^s + \cos(\delta)\sin(\alpha)\,\boldsymbol{h} + \sin(\alpha)\sin(\delta)\,\boldsymbol{q}. \tag{2.1}$$

2.1.2. Optical pressure model

Solar sail propulsion depends on momentum transfer from photons (Fig. 2). In our calculations, we assume purely specular reflection, a perfectly opaque, flat, and unchanging sail, and we do not take into account effects of heat radiation from the sail or effects from shadowing (including from self-shadowing or other celestial bodies) [31,32].

We can relate the known solar irradiance I_E at 1 AU from the sun (energy per unit area and unit time) to momentum through the relationship $E = Mc$ (where E is energy, M is momentum, and c is the speed of light). Using the fact that $I_E = \frac{\Delta E}{A_\alpha \Delta t}$ (where A_α is the area normal to the radiation),

$$I_E = \frac{c\Delta M}{A_\alpha \Delta t} \tag{2.2}$$

By rearranging, we can calculate the solar radiation pressure at 1 AU from the sun (P_E).

$$\frac{I_E}{c} = \frac{\Delta M}{A_\alpha \Delta t} = P_E \tag{2.3}$$

Taking into account the decay of light intensity $\propto \frac{1}{\mathcal{L}^2}$, where $R_E = 1$ AU, and

$$\mathcal{L} = \frac{||\boldsymbol{r}^s||}{R_E} \tag{2.4}$$

we can calculate the solar radiation pressure, P, at a distance of $||\boldsymbol{r}^s||$:

$$P = \frac{P_E}{\mathcal{L}^2} \tag{2.5}$$

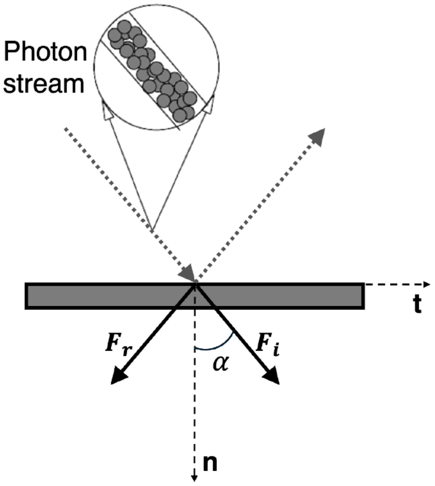

Fig. 2. Force from incident and reflected photons.

The resulting force from incident photons, $\boldsymbol{F}_i$, is

$$\boldsymbol{F}_i = \frac{P_E}{\mathcal{L}^2} A_\alpha \cdot \boldsymbol{u}^s \tag{2.6}$$

and can be written in terms of the normal and transverse direction vectors as:

$$\boldsymbol{F}_i = \frac{P_E}{\mathcal{L}^2} A_\alpha \cdot [\cos(\alpha)\boldsymbol{n} + \sin(\alpha)\boldsymbol{t}] \tag{2.7}$$

The force from reflected photons, $\boldsymbol{F}_r$, can then be written in terms of the reflectivity of the sail, $0 \leq R_{ef} \leq 1$ ($R_{ef} = 0$ represents a scenario with complete absorbance and $R_{ef} = 1$ represents complete reflection).

$$\boldsymbol{F}_r = \frac{P_E}{\mathcal{L}^2} A_\alpha \cdot \left[R_{ef}\cos(\alpha)\boldsymbol{n} - R_{ef}\sin(\alpha)\boldsymbol{t}\right] \tag{2.8}$$

Adding together the force from the incident and reflected photons, the resultant force on the sail is

$$\boldsymbol{F}_s = \frac{P_E}{\mathcal{L}^2} A_\alpha \cdot \left[(1 + R_{ef})\cos(\alpha)\boldsymbol{n} + (1 - R_{ef})\sin(\alpha)\boldsymbol{t}\right] \tag{2.9}$$

Or in terms of the area of a solar sail, A_o, where $A_\alpha = A_o\cos(\alpha)$,

$$\boldsymbol{F}_s = \frac{P_E}{\mathcal{L}^2} A_o\cos(\alpha) \cdot \left[(1 + R_{ef})\cos(\alpha)\boldsymbol{n} + (1 - R_{ef})\sin(\alpha)\boldsymbol{t}\right] \tag{2.10}$$

2.1.3. *Governing equations*

Recall that the *lightness number*, β, is defined as the ratio of the solar radiation pressure acceleration to the solar gravitational acceleration:

$$\beta = \frac{2A_o P_E R_E^2}{G m_{Sun} m_s}. \tag{2.11}$$

We can now write the force of the optical pressure in terms of β (a known property of the solar sail) as

$$\boldsymbol{F}_s = \frac{\beta}{2} \frac{G m_{Sun} m_s}{||\boldsymbol{r}^s||^2} \left[\cos^2(\alpha)(1 + R_{ef})\,\boldsymbol{n} + \cos(\alpha)\sin(\alpha)(1 - R_{ef})\,\boldsymbol{t}\right]. \tag{2.12}$$

Adding in gravitational effects from the Sun, a simplified governing equation for the dynamics of a solar sail is given by:

$$m\ddot{\boldsymbol{r}}^s = -\frac{G m_{Sun} m_s}{||\boldsymbol{r}^s||^2}\boldsymbol{u}^s + \boldsymbol{F}_s(t), \tag{2.13}$$

where G is the universal gravitational constant, m_s is the mass of the solar sail, and m_{Sun} is the mass of the Sun. We can add additional gravitational effects from other celestial bodies to get the final governing equation form, where celestial body i has mass m_i and is located by position vector $\boldsymbol{r}^i$.

$$m\ddot{\boldsymbol{r}}^s = m\dot{\boldsymbol{v}}^s = -\frac{G m_{Sun} m_s}{||\boldsymbol{r}^s||^2}\boldsymbol{u}^s - \sum_{i=1}^{k} \frac{G m_i m_s}{||\boldsymbol{r}^i - \boldsymbol{r}^s||^3}(\boldsymbol{r}^i - \boldsymbol{r}^s) + \boldsymbol{F}_s(t) = \mathcal{F}(t) \tag{2.14}$$

In Cases 1 & 3, only Earth is included as an additional celestial body, since the solar sails remain close to Earth, while in Case 2, the celestial bodies included are Mercury, Venus, Earth, Mars, Jupiter, Saturn, Uranus, and Neptune.

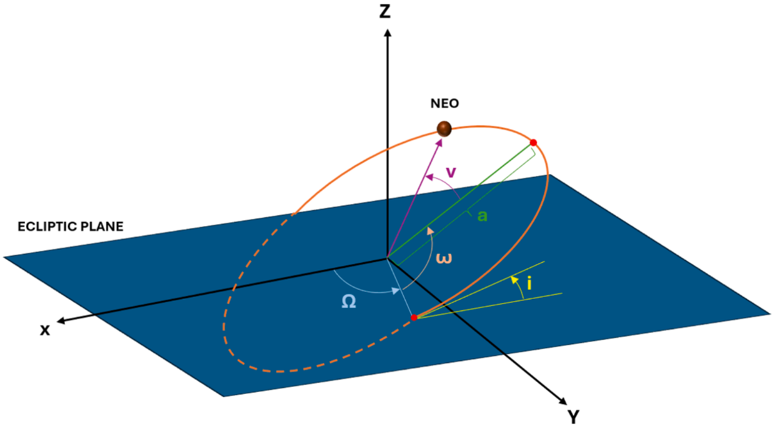

Fig. 3. A visualization of the Keplarian orbital elements.

2.1.4. Numerical solution

The numerical solution of Eq. (2.14) proceeds with an explicit forward-Euler two-step solution, where

$$\boldsymbol{v}^s(t+\Delta t) = \boldsymbol{v}^s(t) + \frac{\Delta t}{m_s}\mathcal{F}(t), \tag{2.15}$$

and

$$\boldsymbol{r}^s(t+\Delta t) = \boldsymbol{r}^s(t) + \boldsymbol{v}^s(t)\Delta t. \tag{2.16}$$

Separately from sail dynamics calculations, we solve for each celestial body's (e.g., planets and NEOs) position at each time step using Kepler's Laws of planetary motion (which represent the solution of a two-body problem) and the six classical orbital elements (length of the semi major axis, a, eccentricity, e, inclination, i, right ascention of the ascending node, Ω, argument of periapsis, ω, and true anomaly, v) as shown in Fig. 3. The x_1 , x_2 , x_3 coordinates of the body are thus calculated as

$$r = \frac{a(1-e^2)}{1+e\cos(v)} \tag{2.17}$$

$$x_1 = r[\cos(\Omega)\cos(\omega+v) - \sin(\Omega)\sin(\omega+v)\cos(i)] \tag{2.18}$$

$$x_2 = r[\sin(\Omega)\cos(\omega+v) + \cos(\Omega)\sin(\omega+v)\cos(i)] \tag{2.19}$$

$$x_3 = r\sin(\omega+v)\sin(i) \tag{2.20}$$

We use the data for each body with an arbitrary starting date of July 12, 2024. The orbital elements for each celestial body were acquired from the JPL Horizon Systems Application (https://ssd.jpl.nasa.gov/horizons/app.html/). We solve for all positions of the celestial bodies of interest (based on initial orbital element values) before we run the sail dynamics model. This allows our framework to be trivially modified to calculate the positions of each celestial body with more precise calculation models (such as NASA's SPICE toolkit).

2.1.5. Network communication modeling

In Case 3, we aim to use a network of solar sails to send collected data back to Earth, assuming that within a communication distance d_c of each other, sails will be able to transfer data. To determine if data from a NEO visit has been sent to Earth, we first identify every "connection" – connections occur when sails/Earth are within distance d_c of each other and are active (sails are active after a delay time of t_{d1} from initial launch, Earth is always active). Next, we traverse through the connections recursively to build a tree, looking first for connections to Earth, then looking for connections to those sails that had connections to Earth, and so on. A detailed description of the connection algorithm is found in the appendix.

Additionally, an optional interpolation step can be used to better identify connections when using coarse time steps. Rather than using the raw data of sail and Earth positions over time, this setting allows the user to perform a linear interpolation between each known position of the objects on a finer time stepping mesh defined by the user. Connections algorithms will then use this approximate position data from the finer mesh.

2.2. Trajectory design space

In our navigation strategy design, the sail position is controlled through design variables containing the length of a time segment and the corresponding position of the sail (defined using cone and clock angles). This results in a step function of positions with variable time segment lengths. Optionally, polynomial interpolation can be used to smooth the sail position function. This set-up enables flexible control while limiting the design space variables, which is important for optimization efficiency.

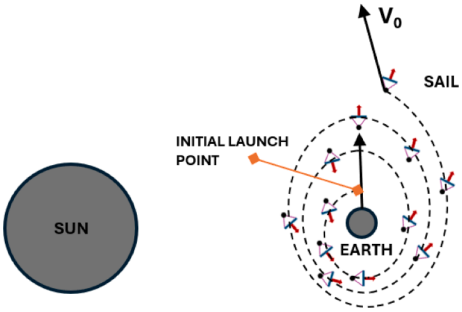

Fig. 4. A schematic of the spiral out maneuver of a solar sail to escape Earth's gravitational well. The red arrow represents the sail's normal, and the different sail locations delineate the progressive change in its orientation to gain more momentum and spiral out. This figure is not drawn to scale, but is rather meant to visually clarify the concept of the maneuver. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

2.2.1. Design parameters

We represent the solar sail trajectory design space for a single sail by the following N design parameters:

$$\begin{aligned}\boldsymbol{\Lambda}^{(\mathbf{i})} &\equiv \{\zeta_1, \dots, \zeta_N\} \\ &\equiv \{D, t_L, S_1, \dots, S_M, \delta_1, \dots, \delta_M, \alpha_1, \dots, \alpha_M\}\end{aligned} \tag{2.21}$$

where D is an interpolation degree for angle shape functions (discussed in detail in this section), t_L is the launch time, M is the number of time segments for the clock and cone angles, $S_1, \dots, S_M$ are the lengths of each time segment, $\delta_1, \dots, \delta_M$ are the values of the clock angles at each time segment, and $\alpha_1, \dots, \alpha_M$ are the values of the cone angles at each time segment. For Case 1 (single solar sail), $\boldsymbol{\Lambda}^{(\mathbf{i})}$ (Eq. (2.21)) fully defines the sail path. For Cases 2 and 3 (n sails), one design contains n single-sail designs: $[\boldsymbol{\Lambda}^{(\mathbf{1})}, \boldsymbol{\Lambda}^{(\mathbf{2})}, \dots, \boldsymbol{\Lambda}^{(\mathbf{n})}]$ — otherwise, the strategy approach is handled similarly as in Case 1.

A sail's trajectory is also influenced by the launch velocity and location. Since directly launching into the desired path out of Earth's orbit may be prohibitively expensive, it is more likely that the solar sail is launched into an orbit path around Earth (for example, geostationary Earth orbit) in conjunction with a satellite or other missions. From a satellite orbit, the solar sail can then follow the modified orbit steering law (MORSL), discussed in McInnes [31] and represented in Fig. 4, to gradually modify its cone angle to spiral out of Earth's gravitational well. We use the initial velocity, V_0, at time $t = t_L$ as the starting point for the simulation of sail dynamics (reflecting the initial velocity within the simulation, not necessarily the mission launch velocity). The maneuver required to reach this point is well studied [16] and is beyond the scope of our simulations. The initial simulation velocity magnitude is equal to the Earth's approximate orbital speed around the Sun ($V_0 \approx 30$ km/s) in the direction tangent to Earth's orbit. The initial simulation location is 10^6 km away from Earth's surface along the direction tangent to Earth's orbit.

We construct the clock and cone angles for a navigation strategy using the aforementioned design parameters through the following procedure:

1. Set the number of segments M to a desired value.
2. Choose an interpolation method for clock and cone angles: piecewise constant or polynomial interpolation.
3. If piecewise constant clock and cone angles are selected, use the constant values α_i and δ_i during segment i (green line in Fig. 5). The length of segment i corresponds to S_i.
4. If polynomial interpolation of clock and cone angles is selected, fit a Lagrange polynomial of degree D for each $D + 1$ clock and cone angles (α_i and δ_i) in the total number of segments M (red line in Fig. 5). If M is not a multiple of $D + 1$ then fit a Lagrange polynomial of degree equal to the remaining points minus 1.
5. Optimize the design parameters for a specific objective, and plug in the optimal values of D, t_L, $S_1, \dots, S_M$, $\alpha_1, \dots, \alpha_M$, and $\delta_1, \dots, \delta_M$ to generate the corresponding clock and cone angles.

A piecewise Lagrange polynomial is a suitable representation for the angle variation over time because there is no requirement for a completely smooth function of angles over time. BLISS can modify its clock and cone angles by 40 degrees in around 15 minutes, which can be approximated by an instantaneous change with respect to the time scale of years of flight. The algorithmic details of this implementation are found in the appendix.

2.2.2. Design parameter limits

The clock angle can theoretically take any value, and we thus constrain it to $\delta \in [-180°, 180°]$. The cone angle, on the other hand, is constrained to $\alpha \in [-70°, 70°]$ to maintain control authority at all times, since if $\alpha \notin (-90°, 90°)$, the Sun can no longer apply an optical pressure to the reflective side of the sail. The length of each time segment is $S_i \in [1, S_{\max}]$, where $S_{\max}$ is a parameter set by the user. The launch time, t_L, is constrained to be within one year from simulation start. M, $S_{\max}$, the total simulation time, and the launch period can be tailored to the required mission objective. For Cases 2 and 3 (with multiple sails), the design parameter bounds shown apply to all sails within the swarm. Table 1 summarizes the values discussed in this section.

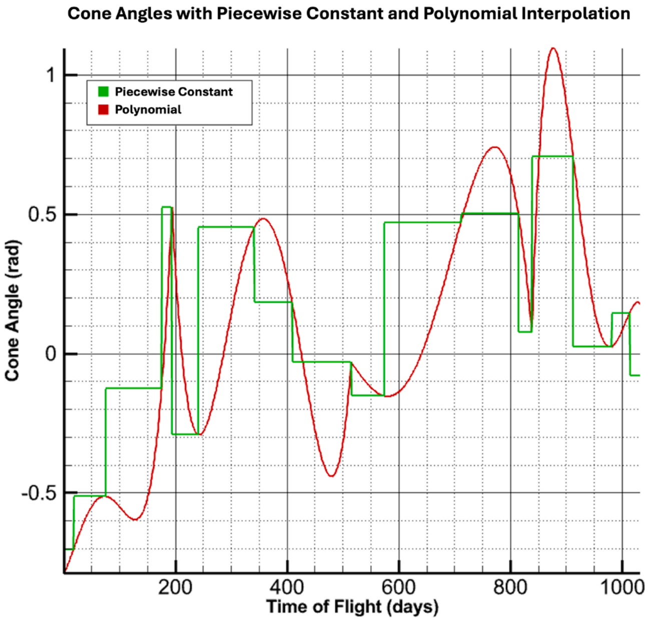

Fig. 5. An example contrasting cone angles with piecewise constant (green) and polynomial interpolation (red) over a time period. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 1
Design Parameters Bounds.

Symbol	Units	Value	Description
$\delta_1, \dots \delta_M$	degree	$[-180, 180]$	Clock angle bounds
$\alpha_1, \dots \alpha_M$	degree	$[-70, 70]$	Cone angle bounds
$S_1, \dots, S_M$	days	$[1, S_{\max}]$	Time segment length bounds
D	none	$[1, 4]$	Interpolation degree
t_L	days	$[1, 365]$	Launch time

2.3. *Optimization approach*

2.3.1. *Design fitness*

The N design parameters constitute a design space that can be explored to optimize for mission objectives. The objective is to optimally control $\boldsymbol{F}_s(t, \alpha, \delta)$ to minimize the following multi-objective cost functions, which correspond to Cases 1, 2, and 3:

$$\begin{aligned}\Pi^{\text{NEO Rendezvous}} &= W_1\Pi^{\text{Sun Penalty}} + W_2\Pi^{\text{Tracking 1}} + W_3\Pi^{\text{Tracking 2}} \\ &\quad + W_4\Pi^{\text{Return}} + W_5\Pi^{\text{Energy}} + W_6\Pi^{\text{Approach Vel.}}\end{aligned} \tag{2.22}$$

$$\begin{aligned}\Pi^{\text{Solar Escape}} &= W_1\Pi^{\text{Sun Penalty}} + W_2\Pi^{\text{Sun Dist.}} + W_3\Pi^{\text{Sail Dist.}} \\ &\quad + W_4\Pi^{\text{Percent Success}}\end{aligned} \tag{2.23}$$

$$\Pi^{\text{Network}} = \begin{cases} \Pi^{\text{End Goal}} & \text{if all info from NEOs is sent back to Earth} \\ \Pi^{\text{End Goal}}[W_2\Pi^{\text{Connections}} + W_3\Pi^{\text{Earth Dist.}} & \\ +W_4\Pi^{\text{Min Dist. to Data}} + W_5\Pi^{\text{Sail Dist. Sum}} & \\ +W_6\Pi^{\text{MinMax Sail Dist.}} + W_7\Pi^{\text{Sun Penalty}}] & \text{otherwise.} \end{cases} \tag{2.24}$$

where each W_i refers to a weight for its respective cost component, which can be varied to place more or less importance on each component.

Case 1 objective function: Eq. (2.22) represents the cost function for Case 1, a sail rendezvous with a NEO followed by a return to Earth. Below, we define the cost components for this mission objective.

- $\Pi^{\text{Sun Penalty}}$: Applies penalty (ξ) if the sail comes within 0.25 AU of the Sun. Recall that the Sun is at the origin of the coordinate system, such that $\|\boldsymbol{r}_t^s\|$ is the distance from the sail to the Sun at time t.

$$\Pi^{\text{Sun Penalty}} = \begin{cases} \xi & \text{if } \min_{t>t_L}(\|\boldsymbol{r}_t^s\|) \leq 0.25 \text{ AU} \\ 0 & \text{otherwise.} \end{cases} \tag{2.25}$$

- $\Pi^{\text{Tracking 1}}$: Incentivizes the sail to "hover" within a distance of TOL_{NEO} (the NEO-sail distance required to rendezvous) to the target NEO. If it reaches that distance, incentivizes the length of time within TOL_{NEO}, H_t, to be close to a desired hover time, $H_{t,des}$. t^* is the time at which the NEO-sail distance was minimized.

$$\Pi^{\text{Tracking 1}} = \begin{cases} |(H_{t,des} - H_t)/H_{t,des}| & \text{if } \|\boldsymbol{r}_{t^*}^s - \boldsymbol{r}_{t^*}^{NEO}\| \leq \text{TOL}_{\text{NEO}} \\ \|\boldsymbol{r}_{t^*}^s - \boldsymbol{r}_{t^*}^{NEO}\| * \xi & \text{otherwise.} \end{cases} \tag{2.26}$$

- $\Pi^{\text{Tracking 2}}$: Similar to $\Pi^{\text{Tracking 1}}$, incentivizes sail to hover near NEO, but with a more relaxed tolerance, TOL_{NEO2}. Having both cost components allows for a more gradual decrease of cost during optimization.

$$\Pi^{\text{Tracking 2}} = \begin{cases} |(2 \cdot H_{t,des} - H_{t2})/(2 \cdot H_{t,des})| & \text{if } \|\boldsymbol{r}_{t^*}^s - \boldsymbol{r}_{t^*}^{NEO}\| \leq \text{TOL}_{\text{NEO2}} \\ \|\boldsymbol{r}_{t^*}^s - \boldsymbol{r}_{t^*}^{NEO}\| * \xi & \text{otherwise.} \end{cases} \tag{2.27}$$

where H_{t2} is the time where the NEO-sail distance is less than TOL_{NEO2} within the time interval defined by the range of time step indices: [max(0, round(min_dist_index – $0.75 \cdot 2 \cdot H_t/dT$)), min(round(min_dist_index – $0.75 \cdot 2 \cdot H_t/dT$))]. min_dist_index is the time step index of time of minimum NEO-sail distance, t^*.

- Π^{Return}: Incentivizes return to Earth by penalizing the final Earth-sail distance if the desired tolerance for return to Earth (TOL_{E}) or NEO rendezvous (TOL_{NEO}) has not been reached. $\boldsymbol{r}_T^E$ and $\boldsymbol{r}_T^s$ refer to the respective locations of Earth and the solar sail at simulation end time.

$$\Pi^{\text{Return}} = \begin{cases} \|\boldsymbol{r}_T^E - \boldsymbol{r}_T^s\| & \text{if } \|\boldsymbol{r}_T^E - \boldsymbol{r}_T^s\| > \text{TOL}_{\text{E}} \text{ and} \|\boldsymbol{r}_{t^*}^s - \boldsymbol{r}_{t^*}^{NEO}\| > \text{TOL}_{\text{NEO}}, \\ 0 & \text{otherwise.} \end{cases} \tag{2.28}$$

- Π^{Energy}: Incentivizes minimized kinetic energy during the sail flight to encourage an energy efficient path. If the sail does not reach the NEO within a tolerance TOL_{NEO4}, the penalty is on energy during the approach towards the NEO (with penalty term ξ), otherwise, it minimizes energy during the return to Earth (with penalty term ψ). Loosely, this variable objective aims to minimize the kinetic energy sequentially over optimization iterations. Penalty terms ξ and ψ are used to adjust the relative emphasis on approach and return energy.

$$\Pi^{\text{Energy}} = \begin{cases} \xi \cdot \sum_{i=0}^{t^*} 0.5 m_s \|\boldsymbol{v}_i^s\|^2 & \text{if } \|\boldsymbol{r}_{t^*}^{NEO} - \boldsymbol{r}_{t*}^s\| > \text{TOL}_{\text{NEO4}}, \\ \psi \cdot \sum_{i=t^*}^{T} 0.5 m_s \|\boldsymbol{v}_i^s\|^2 & \text{otherwise.} \end{cases} \tag{2.29}$$

- $\Pi^{\text{Approach Vel.}}$: Incentivizes a close velocity match (small relative velocity) during rendezvous between the NEO and the solar sail for better imaging.

$$\Pi^{\text{Approach Vel.}} = \begin{cases} \|\boldsymbol{v}_{t^*}^{NEO} - \boldsymbol{v}_{t^*}^s\| & \text{if } \|\boldsymbol{r}_{t^*}^{NEO} - \boldsymbol{r}_{t^*}^s\| > \text{TOL}_{\text{NEO}}, \\ 0 & \text{otherwise.} \end{cases} \tag{2.30}$$

where $\boldsymbol{v}_{t^*}^{NEO}$ and $\boldsymbol{v}_{t^*}^s$ are the velocities of the NEO and the sail, respectively, at time t^*.

Case 2 objective function: Eq. (2.23) depicts the cost function for Case 2, solar system escape using a swarm of solar sails. Sails are indexed by $i = 1, \ldots, n$, where $\boldsymbol{r}_t^{s_i}$ is the location vector for sail i at time t. The cost components for this mission objective are:

- $\Pi^{\text{Sun Penalty}}$: Applies penalty (ξ) if any sails come within 0.2 AU of the Sun.

$$\Pi^{\text{Sun Penalty}} = \begin{cases} \xi & \text{if } \min_{i,\ t>t_l}(\|\boldsymbol{r}_t^{s_i}\|) \leq 0.2 \text{ AU} \\ 0 & \text{otherwise.} \end{cases} \tag{2.31}$$

- $\Pi^{\text{Sun Dist.}}$: Incentivizes sails to be a desired distance, D_{des}, from the sun at simulation end time.

$$\Pi^{\text{Sun Dist.}} = \max_i \left(\left| \frac{D_{des} - \|\boldsymbol{r}_T^{s_i}\|}{D_{des}} \right| \right) \tag{2.32}$$

- $\Pi^{\text{Sail Dist.}}$: Incentivizes sails to be far from each other at simulation end time.

$$\Pi^{\text{Sail Dist.}} = -\min_{i,j,i\neq j} \left(||\boldsymbol{r}_T^{s_i} - \boldsymbol{r}_T^{s_j}|| \right) \tag{2.33}$$

- $\Pi^{\text{Percent Success}}$: Incentivizes a larger fraction of sails to get close to the desired final Sun-sail distance. This term is intended to give a smoother cost variation by incorporating distance information from all sails, rather than just the extremes as in $\Pi^{\text{Sun Dist.}}$.

$$\Pi^{\text{Sail Dist.}} = 1 - \frac{1}{n} \sum_i \begin{cases} 1 & \text{if } \left| \frac{D_{des} - \|\boldsymbol{r}_T^{s_i}\|}{D_{des}} \right| \leq 0.8 \\ 0 & \text{otherwise.} \end{cases} \tag{2.34}$$

Case 3 objective function: Lastly, Eq. (2.24) represents the cost function for Case 3, a sail network in which one or more sails rendezvous with a NEO and returns the collected data to Earth via communication with a network of sails.

For Case 3, we first optimize a sail (or multiple sails) using Case 1's cost function (Eq. (2.23)) with $W_4 = 0$ (weight on the cost for return to Earth) to get sail trajectories that visit a NEO but are not optimized to return to Earth. The trajectory of this pre-optimized sail (or if multiple, these sails) is then locked in, not to be further optimized, while additional network sails are optimized according to cost function Eq. (2.24) to return the collected NEO data back to Earth.

The cost components for Case 3 are:

- $\Pi^{\text{End Goal}}$: incentivizes the goal of returning information from NEO visits to Earth in minimal time. We consider a case where sails make $k = 1, 2, \dots, q$ visits to NEOs. The total time to return information from NEO visit k to Earth is I_r^k, where total time is measured from the beginning of the simulation.

$$\Pi^{\text{End Goal}} = \prod_{k=1} \begin{cases} \frac{1}{W_1} \cdot \frac{I_r^k}{T} & \text{if data from visit } k \text{ is returned to Earth,} \\ 1 & \text{otherwise.} \end{cases} \tag{2.35}$$

- $\Pi^{\text{Connections}}$: Incentivizes a larger number of connections in the communication tree, which loosely represents the total amount of productive communication. The connections (instances of sails/Earth being within communication distance, d_c, of each other) in the communication tree chain together to relay data back to Earth (see the section communication-modelingNetwork Communication Modeling and the appendix for further detail). The number of connections in the communication tree is represented as n_c.

$$\Pi^{\text{Connections}} = \frac{1}{n_c + 1} \tag{2.36}$$

- $\Pi^{\text{Earth Dist.}}$: Incentivizes the minimum distance between any sail and Earth to be small, measured after a delay time of t_{d2} from initial launch to encourage sail-Earth communication.

$$\Pi^{\text{Earth Dist.}} = \min_{i,\ t > t_L + t_{d2}} (\|\boldsymbol{r}_t^E - \boldsymbol{r}_t^{s_i}\|) \tag{2.37}$$

- $\Pi^{\text{Min Dist. to Data}}$: Incentivizes a small minimum distance between a sail that has visited a target NEO and any other sail. If there are multiple sails visiting NEOs or sails that perform multiple visits ($k > 1$), the minimum distances are summed across visit instances. The time of NEO visit k is t_k. Take the index of the sail that performed a given visit k to be j, where the indices of all pre-optimized sails visiting NEOs is contained in set V.

$$\Pi^{\text{Min Dist. to Data}} = \sum_k \left(\min_{i \neq j,\ i \notin V,\ t > t_k} (\|\boldsymbol{r}_t^{s_i} - \boldsymbol{r}_t^{s_j}\|) \right) \tag{2.38}$$

- $\Pi^{\text{Sail Dist. Sum}}$: Incentivizes sails to have a small minimum distance to any other sail. The minimum distances are calculated after the time of the first NEO visit, $\min_k t_k$.

$$\Pi^{\text{Sail Dist. Sum}} = \sum_i \left(\min_{j \neq i,\ t > \min_k t_k} (\|\boldsymbol{r}_t^{s_i} - \boldsymbol{r}_t^{s_j}\|) \right) \tag{2.39}$$

- $\Pi^{\text{MinMax Sail Dist.}}$: Incentivizes sails to have a large maximum distance from other sails. Again, distances are calculated after the time of the first NEO visit, $\min_k t_k$.

$$\Pi^{\text{MinMax Sail Dist.}} = \exp\left(-\min_i \left(\max_{j \neq i,\ t > \min_k t_k} (\|\boldsymbol{r}_t^{s_i} - \boldsymbol{r}_t^{s_j}\|) \right) \right) \tag{2.40}$$

- $\Pi^{\text{Sun Penalty}}$: applies penalty (W_7) if any sails come within 0.25 AU of the Sun.

$$\Pi^{\text{Sun Penalty}} = \begin{cases} W_7 & \text{if } \min_{i,\ t > t_l} (\|\boldsymbol{r}_t^{s_i}\|) \leq 0.25 \text{ AU} \\ 0 & \text{otherwise.} \end{cases} \tag{2.41}$$

2.3.2. *Genetic algorithm*

We follow Zohdi [33–36] in order to minimize the cost functions in Eqs. (2.23) and (2.22). The parameters for the GA framework we implement are shown in Table 2, and the algorithm can be summarized as follows:

- **STEP 1:** Generate S genetic strings, with the design parameters for each string randomly sampled from ranges specified by the design parameter limits. This creates a population of genetic strings as:

$$\boldsymbol{\Lambda} = \{\boldsymbol{\Lambda}^{(1)}, \boldsymbol{\Lambda}^{(2)}, \dots, \boldsymbol{\Lambda}^{(\mathbf{S})}\} \tag{2.42}$$

Table 2
Genetic Algorithm Parameters.

Symbol	Value	Description
P	10	Number of parent strings kept after each generation
S	30	Total number of genetic strings per generation
Q	10	Number of eliminated strings in each generation
G	100	Maximum number of generations

where

$$\mathbf{\Lambda^{(i)}} = \begin{pmatrix} D^- \le D^{(i)} \le D^+ \\ t_L^- \le t_L^{(i)} \le t_L^+ \\ S_1^- \le S_1^{(i)} \le S_1^+ \\ \vdots \\ S_M^- \le S_M^{(i)} \le S_M^+ \\ \delta_1^- \le \delta_1^{(i)} \le \delta_1^+ \\ \vdots \\ \delta_M^- \le \delta_M^{(i)} \le \delta_M^+ \\ \alpha_1^- \le \alpha_1^{(i)} \le \alpha_1^+ \\ \vdots \\ \alpha_M^- \le \alpha_M^{(i)} \le \alpha_M^+ \end{pmatrix} \quad \text{for} \quad i = 1, \dots, \mathtt{S}. \tag{2.43}$$

- **STEP 2:** Evaluate the population's fitness by plugging each string into the cost function $\Pi(\mathbf{\Lambda^{(i)}})$ and store the respective cost value.
- **STEP 3:** Rank the performance of each string $\Pi(\mathbf{\Lambda^{(i)}})$ in ascending order, such that the best performers are first.
- **STEP 4:** Pairwise mate the top P (parent) strings to produce children strings as:

$$\begin{cases} \mathbf{\Lambda^{(P+i)}} & = \varphi_{\mathrm{P+i}}\mathbf{\Lambda^{(i)}} + (1 - \varphi_{\mathrm{P+i}})\mathbf{\Lambda^{(i+1)}} \\ \mathbf{\Lambda^{(P+i+1)}} & = \varphi_{\mathrm{P}+i+1}\mathbf{\Lambda^{(i)}} + (1 - \varphi_{\mathrm{P+i+1}})\mathbf{\Lambda^{(i+1)}} \end{cases} \tag{2.44}$$

For $i = 1, 3, 5 \dots \mathtt{P} - 1$, and $\varphi_{\mathtt{P}}, \dots, \varphi_{\mathtt{P+P}} \in Unif(0, 1)$

- **STEP 5:** Eliminate the bottom Q = S - 2P strings and keep top P parent strings and their P children.
- **STEP 6:** Repeat STEPS 1–6 with top gene pool (P parents and P offspring), plus Q new, randomly generated, strings.

2.3.3. Coordinate descent

After applying the GA as a global search algorithm, meant to identify an area of low cost, we then apply a coordinate descent method to optimize within that local low-cost area. An exhaustive review of coordinate descent methods can be found in the texts of Calafiore [37], Bertsekas [38] and Luenberge [39].

Our coordinate descent algorithm is shown in detail in Algorithm 1. This is a simple algorithm that perturbs each of the obtained design variables from the genetic algorithm optimization and evaluates the cost function for one perturbed variable at a time.

3. Results and discussion

3.1. Case 1: round trip mission

In Case 1, a single sail rendezvous with a NEO, then returns to Earth. Since BLISS communicates via laser, data return requires a direct view of Earth and a distance of around 0.1 AU. BLISS also needs to reach a rendezvous distance within 100,000 km of the NEO (depending on its size) to take a picture that shows the NEO in at least one pixel (we later discuss how our framework can be utilized to achieve a rendezvous distance of just 1 km). We select 101955 Bennu as a NEO example for this mission objective. We start the mission on the arbitrary date of July 12, 2024, and we obtain the Keplerian elements of Earth and 101955 Bennu's orbits from the JPL Horizons application. We aim to minimize the objective function Eq. (2.23) subject to the constraints in Table 1.

Fig. 6 shows a 3D visualization of the sail normal variation throughout its optimized flight path. Fig. 7 depicts the evolution of the resulting optimized navigation trajectory, and Table 3 shows the model inputs used to obtain this trajectory. The sail reaches Bennu within the required distance tolerance after 788 days of flight, launches 207 days after simulation start, and returns to Earth after 1577 days of simulation time.

Algorithm 1 Coordinate descent algorithm.

```
Require: design variables (var), maximum iterations, tolerance (tolerance), cost tolerance (cost_tolerance), step size (m)
Evaluate: cost $\leftarrow$ cost_eval(var)
Set no_change $\leftarrow 0$
for j $= 1$ to max iterations do
    for i $= 1$ to length(var) do
        if no_change $>$ tolerance then
            Set the update m $\leftarrow$ m$/2$
            Set no_change $\leftarrow 0$
        end if
        Copy design variables: inc_var $\leftarrow$ var and dec_var $\leftarrow$ var
        Update inc_var[i] $\leftarrow$ inc_var[i] $+$ inc_var[i] $\cdot$ m
        Update dec_var[i] $\leftarrow$ dec_var[i] $-$ dec_var[i] $\cdot$ m
        Evalaute inc_var_cost $\leftarrow$ cost_eval(inc_var)
        Evalaute dec_var_cost $\leftarrow$ cost_eval(dec_var)
        if cost $>$ inc_var_cost then
            Set var $\leftarrow$ inc_var and cost $\leftarrow$ inc_var_cost
            Set no_change $\leftarrow 0$
        else
            Set no_change $\leftarrow$ no_change $+ 1$
        end if
        if cost $>$ dec_var_cost then
            Set var $\leftarrow$ dec_var and cost $\leftarrow$ dec_var_cost
            Set no_change $\leftarrow 0$
        else
            Set no_change $\leftarrow$ no_change $+ 1$
        end if
        if cost $\leq$ cost_tolerance then
            return var
        end if
    end for
end for
return var
```

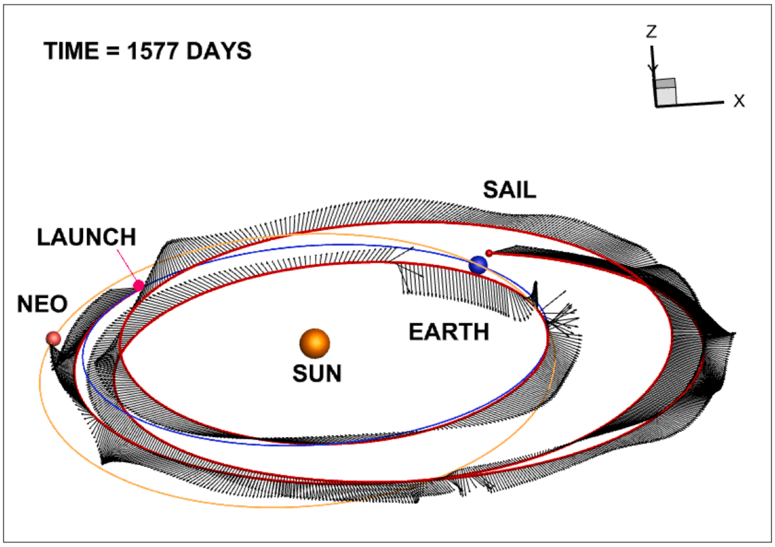

Fig. 6. A 3D visualization of the change in the sail's outward normal direction during its flight path (normal direction is visualized once per day for plot clarity).

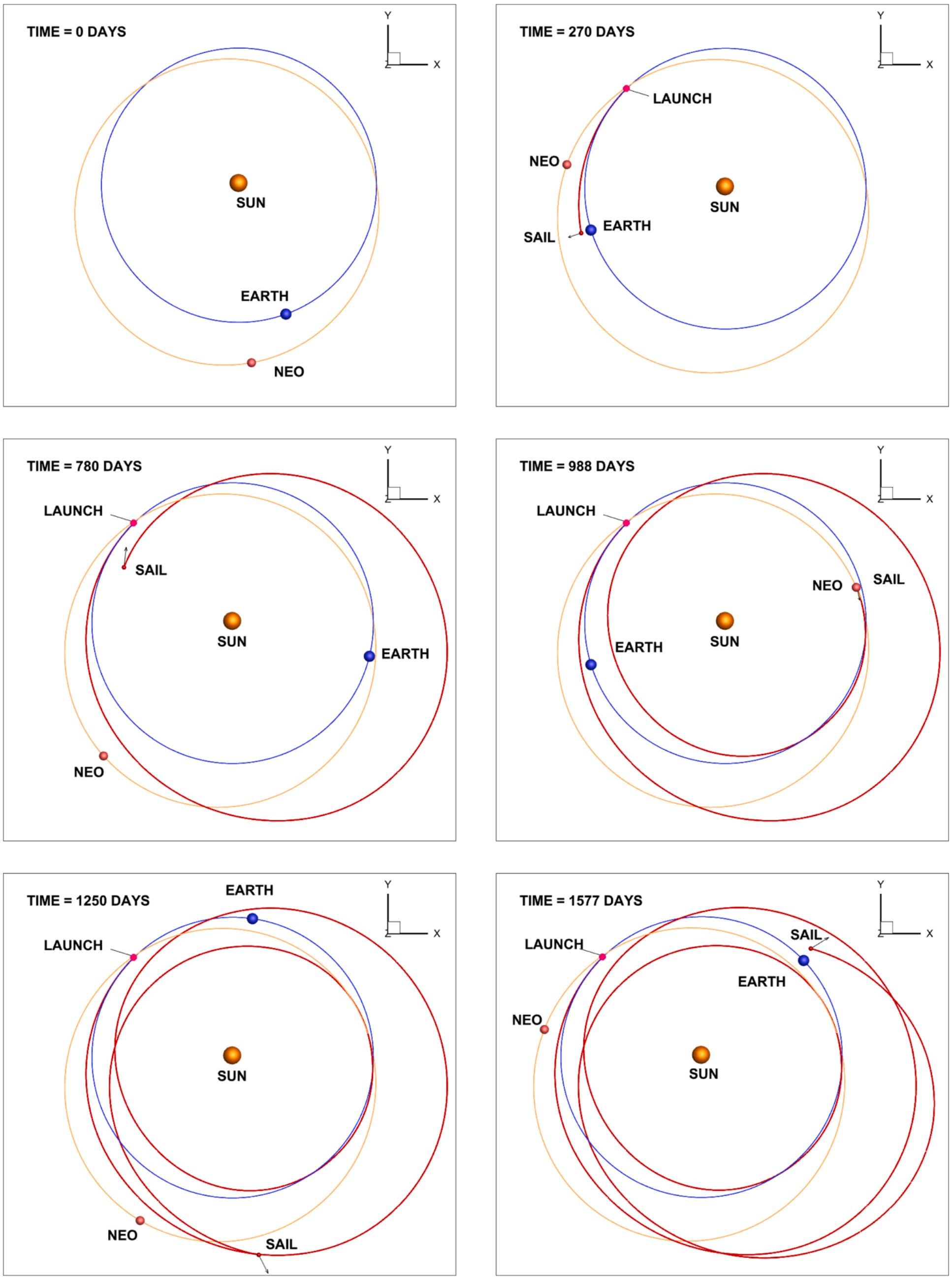

Fig. 7. Snapshots of sequential frames for Case 1 where the solar sail launches, reaches Bennu, and returns to Earth. The arrow pointing out of the sail represents the outward normal to the solar sail. The red line depicts the sail's orbit, the blue line represents Earth's orbit, and the orange line represents Bennu's orbit. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article).

3.2. Case 2: sail swarm solar system escape

The small size and weight of BLISS allows for stacking multiple sails in one launch, which can then act as a space exploration swarm for dispersing magnetometers across and outside the solar system. Fig. 8 shows snapshots of frames depicting the solar system escape trajectories for a swarm of 18 solar sails. These trajectories were again optimized by the genetic algorithm followed by the coordinate descent approach.

These trajectories show an overall successful mission objective, which indicates a well crafted cost function. We see a solar system escape of all sails at simulation end time along with a minimum distance between two sails of 12 AU. The inputs for generating this model are shown in Table 4. This can be trivially scaled to a larger number of sails, albeit at the expense of more computational time due to the larger number of variables to optimize with the coordinate descent approach.

Table 3
Case 1 Model Input Values.

Symbol	Value	unit	Description
W_1	1	N/A	Sun penalty cost weight
W_2	10	N/A	Tracking 1 cost weight
W_3	10	N/A	Tracking 2 cost weight
W_4	1	N/A	Return cost weight
W_5	10^{11}	N/A	Energy cost weight
W_6	10	N/A	Approach velocity cost weight
ξ	10^5	N/A	General penalty term
ψ	10^2	N/A	Return energy penalty term
$\mathrm{TOL}_{\mathrm{NEO}}$	10^5	km	Tolerance for NEO rendezvous
$\mathrm{TOL}_{\mathrm{NEO2}}$	0.05	AU	Additional NEO tolerance value
$H_{t,des}$	80	days	Desired hover time
$S_{\max}$	90	days	Maximum segment length bound
M	40	N/A	Number of time segments
T	5	years	Total simulation time
Δt	0.1	days	Simulation time step

Table 4
Case 2 Model Input Values.

Symbol	Value	unit	Description
W_1	1	N/A	Sun penalty cost weight
W_2	1	N/A	Distance away from the Sun cost weight
W_3	1	N/A	Distances between sails cost weight
W_4	10	N/A	Successful escapes cost weight
D_{des}	100	AU	Desired final Sun-sail distance
ξ	10^5	N/A	Penalty term
$S_{\max}$	500	days	Maximum segment length bound
M	30	N/A	Number of time segments
T	15	years	Total simulation time
Δt	5	days	Simulation time step
N	15	N/A	Number of solar sails

3.3. Case 3: swarm network communication

Case 3 lays out a framework to optimize a network of sails to send data back to Earth from one or multiple NEO visits. We demonstrate this approach for a single NEO visit and a network of 10 sails.

As in Case 1, we target 101955 Bennu as the object of interest, starting with an arbitrary date of July 12, 2024. A single, pre-optimized sail visits Bennu to collect NEO data. This sail trajectory is optimized using the same settings as Case 1, but with the weight on return set to 0. Taking this pre-optimized trajectory as a constant, a network of 10 additional sails is then optimized to return the data to Earth according to Eq. (2.24) and using the weights shown in Table 5. The communication distance d_c, is taken to be 10^7 km, which is larger than the current estimated communication distance of BLISS ($\sim 10^6$ km). Given potential future improvements in the

Table 5
Case 3 model input values.

Symbol	Value	unit	Description
W_1	2	N/A	Cost weight for success and speed of data return
W_2	10	N/A	Cost weight for number of connections in `connectionPaths`
W_3	10^7	N/A	Cost weight for minimum distance between sails and Earth
W_4	10	N/A	Cost weight for minimum distance between sail with collected data and any other sail
W_5	1000	N/A	Cost weight for sum of minimum distances between sails
W_6	10^5	N/A	Cost weight for minimum of maximum of distances between sails
W_7	10^7	N/A	Cost weight for sun penalty
$S_{\max}$	70	days	Maximum segment length bound
M	30	N/A	Number of time segments
T	5	years	Total simulation time
Δt	5	days	Simulation time step
N_o	10	N/A	Number of optimizable solar sails
d_c	10^7	km	Communication distance
t_{d1}	100	days	Time delay after launch for start of communication
t_{d2}	200	days	Time delay after launch used in cost function term
k	1	N/A	Number of NEO visits

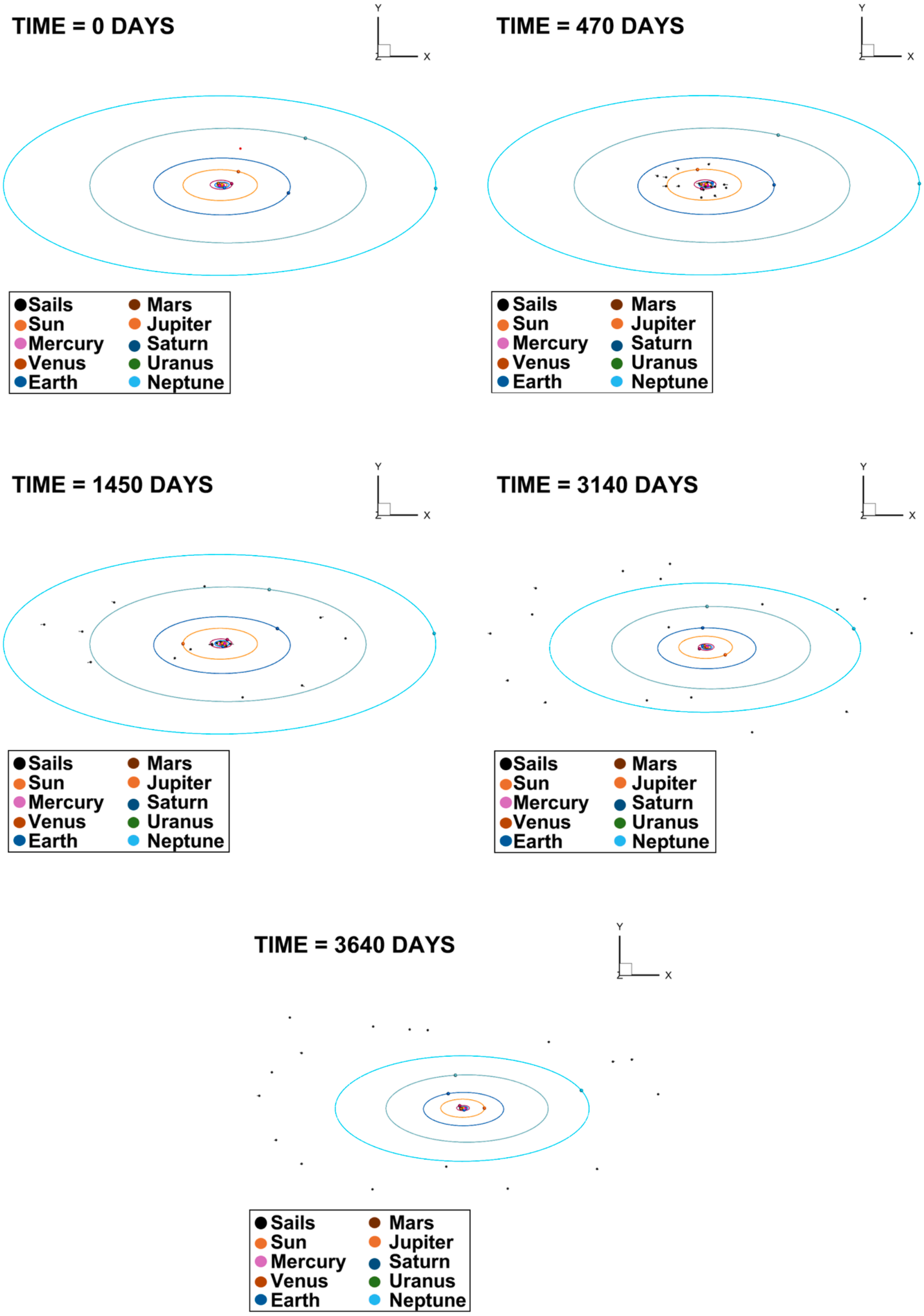

Fig. 8. Snapshots of sequential frames for Case 2 showing the dispersion of a swarm of 18 solar sails over 10 years.

communication distance of BLISS as well as the limited location-identification accuracy of the sails to begin with, we leave further improvement at smaller d_c values to future work.

The resulting trajectories are shown in Fig. 9. We observe a relatively fast return of data to Earth: in Case 1, the sail is able to return data to Earth after 1577 days, where the network of sails is able to return the data from the NEO visit to Earth after just 510 days. Since the objective function and communication method are different between Case 1 and 3, it is not immediately clear if the time improvement is due to the swarm structure, or to another change.

To disentangle the effects of these changes, we first go through the relevant changes to the objective function and then to the observed differences in results. First, we note that there is a greater emphasis on speed of return in the cost function for Case 3

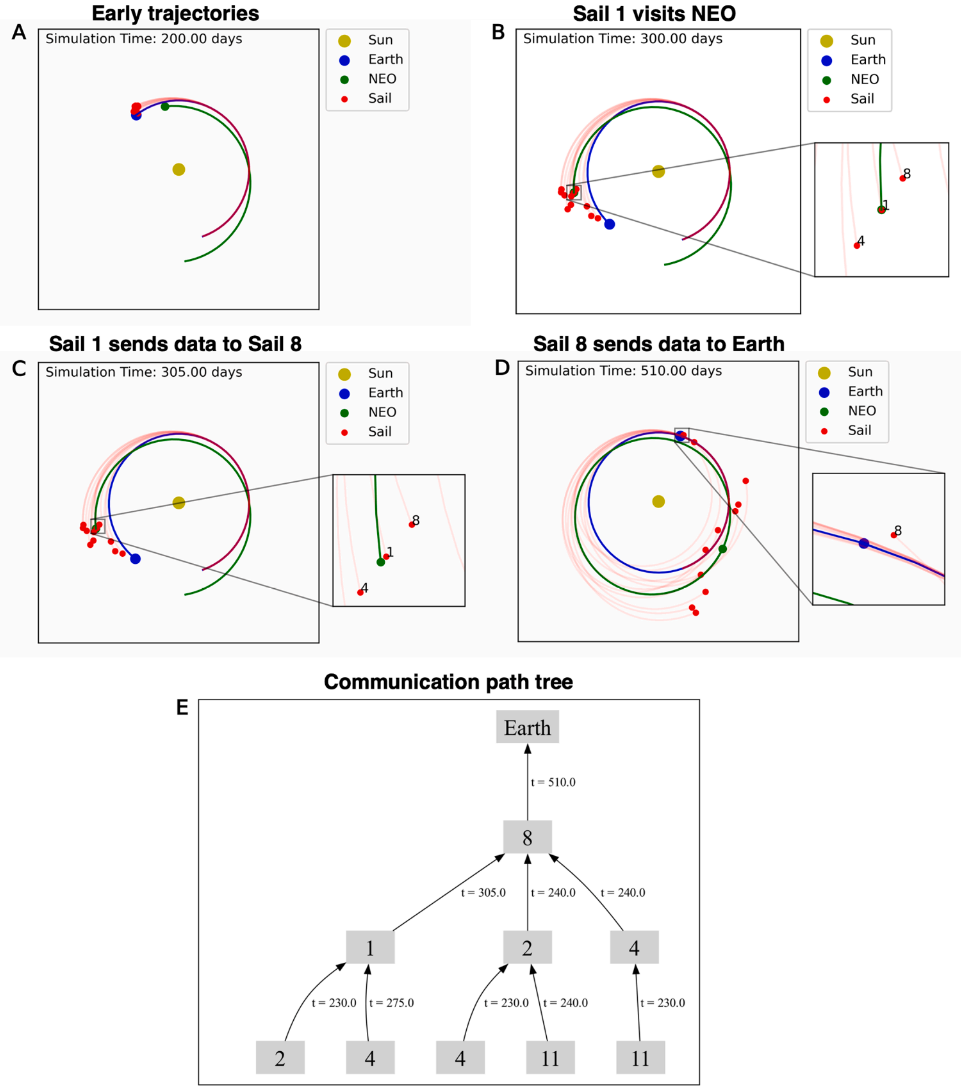

Fig. 9. Example of the trajectories of a network of sails optimized to send data back from NEO visit to Earth. (A) Early trajectories at $t = 200$ days. (B) Trajectories at $t = 300$ days, where Sail 1 visits the NEO and collects data. (C) Trajectories at $t = 305$ days, where Sail 1 is close enough to send data to Sail 8. (D) Trajectories at $t = 510$ days, where Sail 8 sends data back to Earth. (E) Full tree of all data communication paths that send data back to Earth.

Table 6
Model input values for a 1km rendezvous distance.

Symbol	Value	unit	Description
W_1	1	N/A	Sun penalty cost weight
W_2	10	N/A	Tracking 1 cost weight
W_3	10	N/A	Tracking 2 cost weight
W_4	10	N/A	Return cost weight
W_5	10^{11}	N/A	Energy cost weight
W_6	0	N/A	Approach velocity cost weight
ξ	10^9	N/A	General penalty term
ψ	10	N/A	Return energy penalty term
$\mathrm{TOL}_{\mathrm{NEO}}$	1	km	Tolerance for NEO rendezvous
$\mathrm{TOL}_{\mathrm{NEO2}}$	0.05	AU	Additional NEO tolerance value
$H_{t,des}$	80	days	Desired hover time
$S_{\max}$	65	days	Maximum segment length bound
M	55	N/A	Number of time segments
T	5	years	Total simulation time
Δt	0.1	days	Simulation time step

(Eq. (2.24)), where for Case 1, speed of return is only indirectly considered through the "energy" term in the cost function (Eq. (2.22)). We also note that the swarm communication is dependent on the distance d_c for Case 3, where Case 1 does not have any swarm communication and therefore no direct equivalent to d_c. However, the overall distance requirements are similar between the two cases: in Case 1 and Case 3, the NEO rendezvous distances are both 10^5 km, while the Earth return distance is slightly smaller in Case 3 at $d_c = 10^7$ km compared to 0.1 AU $\approx 1.5 \times 10^7$ km in Case 1. Looking at the optimized paths, we observe that the sail pre-optimized to visit Bennu for Case 3 reaches the NEO after 305 simulation days – this sail is not required to return to Earth after the NEO visit (recall that the pre-optimized sail trajectory is determined using the Case 1 cost function with the weight on return to Earth set to 0). In comparison, the sail in Case 1, which *is* required to return to Earth, takes 995 simulation days to initially reach Bennu. All other cost weights and requirements are held the same between these two cases: removing the requirement of return to Earth appears to allow for a faster NEO visit. Therefore, the network structure of the swarm likely helps enable faster return by loosening trajectory requirements on each individual sail.

Fig. 9E visualizes all connections that could lead to data being sent back to Earth. Though the tree does include connections from sails that haven't collected data from a NEO visit, it is not a full tree of all connections (for example, a connection would not be shown for two sails that come within distance d_c of each other, where neither is able to connect back to Earth at a later time). Besides sending NEO data back to Earth, these connections can be used to improve location-identification accuracy by tethering the network back to a known reference location.

4. Conclusions and model extensions

In this paper, we develop a model for individual solar sails and swarms of solar sails and a corresponding optimization method to identify suitable navigation strategies. We demonstrated the capability of this system on three cases: 1.) to coordinate a rendezvous with a NEO and return to Earth, 2.) to coordinate solar system escape for a swarm of sails, and 3.) to achieve swarm network communication for NEO exploration. Using a genetic algorithm and coordinate descent, our framework was able to identify suitable trajectories for each case. Our results demonstrated that swarm communication can significantly speed up exploration and data return, since the network structure allows for looser trajectory requirements on each individual sail. There are, however, multiple extensions and improvements that may be necessary for some mission objectives, so we discuss how our model can handle such extensions.

The discussed rendezvous distance of 100,000 km for a small NEO like Bennu may be too large for BLISS's current capabilities to capture a high-quality image. To achieve a high quality picture (i.e. one where the view of Bennu fills most of the camera sensor area), a much smaller rendezvous distance of 1 km may be necessary. The proposed framework can still achieve this with the parameters shown in Table 6. This resulted in a rendezvous time of 529 days after launch and a total flight time of 1795 days. The ability of our framework to be slightly modified to capture such a small rendezvous distance requirement demonstrates its robustness and flexibility for arbitrary rendezvous distances depending on the size of the NEO of interest. This flexibility is critical for a broader extension that would include generating launch dates and trajectories to get to all of the 1000 NEOs more than 1 km in diameter.

Another important extension is the consideration of the direction of communication. BLISS is intended to communicate to Earth its captured images via a laser source. This signal will be captured on Earth via a telescope pointed in some unit direction at return time, $\boldsymbol{u}_T$, with a specific angular field of view, θ. This direction is tied to Earth's rotation, so a look up value for the direction at the specific time would be necessary. To capture the signal correctly, the laser source must be pointed in a region that falls within the telescope's field of view. This must then be incorporated into the model's cost function. If, for example, the direction of the laser source is aligned with the unit normal direction of the light sail, $\boldsymbol{n}$, then for Case 1, the cost function can have an additional component with a penalty, ξ, at simulation end time T as

$$\Pi^{\text{Signal Dir.}} = \begin{cases} \boldsymbol{n}_T \cdot \boldsymbol{u}_T & \text{if } \boldsymbol{n}_T \cdot \boldsymbol{u}_T < 0 \text{ and } \boldsymbol{n}_T \cdot \boldsymbol{u}_T \geq \cos\frac{\theta}{2} \\ \xi & \text{otherwise.} \end{cases} \tag{4.1}$$

A similar argument can also be made for network communication. When each sail is within communication distance, an extra cost function component similar to that of Eq. (4.1) can be added to the cost to ensure a correct communication direction between two sails at the desired time is achieved.

While this framework handles navigation optimization, practical implementation will require additional real-time correction and control strategies based on the solar sail's information about its location in space. It would also require careful design of a safe mode for the spacecraft, since thrust cannot simply be "turned off" as in other spacecraft missions, and there is often a fairly small window of trajectories that will maintain a feasible path to the desired target [21]. A possible future extension is to use our framework to power a target-identification algorithm, which would be able to find feasible targets for a given launch time and provide backup targets in the case that the original trajectory is missed.

The model could also be extended with more detailed modeling of solar radiation pressure processes to account for things like imperfect reflection, shadowing, and changing optical parameters [40]. Additionally, future models may take into consideration the location of the sun relative to the sail and target of interest (for example, to avoid pointing cameras directly at the sun). As the ideal use-cases for solar sails are long-duration, high-energy missions, future work could also aim to develop even further in that direction, for example with swarm networks that visit *multiple* NEOs while returning data intermittently, or with even longer distance rendezvous-and-return missions.

CRediT authorship contribution statement

Mostafa Sedky: Writing – original draft, Investigation, Formal analysis, Conceptualization; **Maya Horii:** Writing – original draft, Investigation, Formal analysis, Conceptualization; **Thomas Hosmer:** Writing – original draft, Investigation, Formal analysis, Conceptualization; **Kristofer Pister:** Writing – original draft, Conceptualization; **Tarek Zohdi:** Writing – original draft, Conceptualization.

Data availability

No data was used for the research described in the article.

Declaration of competing interest

The authors declare that they have no conflict of interest.

Appendix A.

A.1. Clock and cone angle interpolation

Algorithm 2 describes a general approach for generating the variable angles, where *lagrangePoly()* is a function that fits a Largange polynomial for angles over time by solving a Vandermonde system of linear equations.

Algorithm 2 Create angle functions.

```
Require: Number of time segments (M), clock_angles, cone_angles, polynomial degree (D)
Ensure: polyClocks, polyCones
Initialize polyClocks and polyCones as empty lists
Set i ← 0
while i < M-1 do
    Set points_remaining ← M - i
    Set fit_degree ← min(D, points_remaining - 1)
    Set points_needed ← fit_degree + 1
    Set clocks ← clock_angles[i:i + points_needed]
    Set cones ← cone_angles[i:i + points_needed]
    Set times ← time_segments[i:i + points_needed]
    Set polyClocks[i] ← lagrangePoly(times, clocks)
    Set polyCones[i] ← lagrangePoly(times, cones)
    Set i ← i + fit_degree
end while
return polyClocks, polyCones
```

A.2. Network communication modeling

Algorithm 3 describes the overall tree traversal procedure. Within this process, Algorithm 4 is first used to find all connections (`unprocessedConnections`), then Algorithm 5 (`findBranches()`) is used recursively to find branching connections to the object with index `newTarget` that occur before `time`, traversing the tree in reverse chronological order.

To determine if data from a NEO visit has been sent to Earth, we first identify every “connection” – connections occur when sails/Earth are within distance d_c of each other and are active (sails are active after a delay time of t_{d1} from initial launch, Earth is always active). This process is represented in Algorithm 4 and results in a set of `unprocessedConnections`.

Next, we traverse through the connections recursively to build a tree, looking first for connections to Earth, then looking for connections to those sails that had connections to Earth, and so on. This tree is represented with the list `connectionPaths`, which contains entries of the form `[target, source, timeOfConnection, pathLoc]`, where `target` is the index of the object (Earth or sail) receiving data, `source` is the index of the object sending data, `timeOfConnection` is the time of the connection, and `pathLoc` is the index (within `connectionPaths`) of the parent connection. Objects are indexed such that Earth always has index 0, followed by sails with indexes ≥ 1. As an example: `connectionPaths = [[0, 1, 5, -1], [0, 3, 10, -1], [3, 1, 2, 1], [3, 2, 7, 1]]` represents the connection paths shown in Fig. 10.

For example, `findBranches()` is first run on `newTarget = 0` (representing Earth with index 0). `findBranches()` checks the connections for which Earth is the recipient of data and then identifies any connections that are before the parent connection in the tree – in this case, being the first node of the tree, there are no parent connections, so connections must only be within the simulation time length.

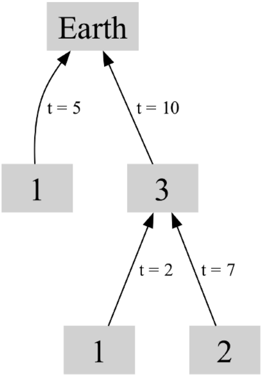

Fig. 10. Example of a tree of data communication paths that send data back to Earth.

For consecutive connections, only the last valid connection is kept (such that `timeOfConnection` is still before the time of the parent connection). Consecutive connections are defined as connections between two objects that occur at every time step during a segment of time. As seen in Algorithm 5, the kept connections are then added to `connectionQueue`. Those connections and any other processed connections are then removed from `unprocessedConnections`. Both directions of connection are removed – that is, both `(i,j,t)` from `unprocessedConnections[i]` and `(j,i,t)` from `unprocessedConnections[j]` – since all incoming connections (`newTarget` as target) have already been identified and any outgoing connections (`newTarget` as source) could only result in the data from the target object being delivered to Earth at the same or a later time.

Algorithm 3 buildConnectionsTree.

```
Require: T (total time of simulation), dT (time step)
Require: objPos (Earth and sail positions, in that order)
dist_matrix, unprocessedConnections = findConnections(objPos)
newTarget = 0 (Initial newTarget is Earth)
pathLoc = -1 (No parent connections, so pathLoc = -1)
time = T
Update unprocessedConnections, connectionQueue with findBranches
Initialize connectionPaths as empty list
while len(connectionQueue) > 0 do
    con = connectionQueue[0]
    (target, source, time, pathLoc) = con
    connectionPaths.append(con)
    pathLoc = len(connectionPaths)-1
    newTarget = source
    Update unprocessedConnections, connectionQueue with findBranches
    connectionQueue.pop(0)
end while
return dist_matrix, connectionPaths
```

Algorithm 4 findConnections.

```
Require: objPos (Earth and sail positions, in that order)
dist_matrix ← array w/ shape (numObjs, numObjs, numTimeSteps)
unprocessedConnections ← dictionary with keys ∈ (0, numObjs)
for i = 0 to numObjs do
    for j = i+1 to numObjs do
        dist ← distances between object i and object j over time
        dist_matrix[i,j,:] ← dist
        for t where dist_matrix[i,j,t] ≤ $d_c$ and both objects are active do
            Append unprocessedConnections[i] with (i,j,t)
            Append unprocessedConnections[j] with (j,i,t)
        end for
    end for
end for
return dist_matrix, unprocessedConnections
```

Algorithm 5 findBranches.

```
Require: connectionQueue, unprocessedConnections, newTarget, pathLoc, time
potentialConnections = unprocessedConnections[newTarget]
lenC = length(potentialConnections)
i = 0
while i < lenC-1 do
    connection = potentialConnections[i]
    nextConnection = potentialConnections[i+1]
    if branchTime $\leq$ time then
        while (nextConnection is consecutive & valid) and (i < lenC-1) do
            Remove connection (both directions) from unprocessedConnections
            i += 1
            connection = potentialConnections[i]
            nextConnection = potentialConnections[i+1]
        end while
        Append connection to connectionQueue
        Remove connection (both directions) from unprocessedConnections
    end if
    i += 1
end while
return unprocessedConnections, connectionQueue
```

References

[1] D. Perna, M.A. Barucci, M. Fulchignoni, et al., The near-Earth objects and their potential threat to our planet, Astron. Astrophys. Rev. 21 (1) (2013) 65. https://doi.org/10.1007/s00159-013-0065-4

[2] N. Anthony, M.R. Emami, Asteroid engineering: the state-of-the-art of near-Earth asteroids science and technology, Prog. Aerosp. Sci. 100 (2018) 1–17. https://doi.org/10.1016/j.paerosci.2018.05.001

[3] E. Vieira Neto, P. Pires, S. Giuliatti Winter, et al., Trajectories for mining space mission on asteroids in near-Earth orbit, Eur. Phys. J. Spec. Top. 232 (18–19) (2023) 2967–2974. https://doi.org/10.1140/epjs/s11734-023-01016-y

[4] R.C. Boden, A.M. Hein, J. Kawaguchi, et al., Target selection and mass estimation for manned NEO exploration using a baseline mission design, Acta Astronaut. 111 (2015) 198–221. https://doi.org/10.1016/j.actaastro.2015.02.018

[5] A.W. Harris, G. D'Abramo, The population of near-Earth asteroids, Icarus 257 (2015) 302–312. https://doi.org/10.1016/j.icarus.2015.05.004

[6] A. Mainzer, T. Grav, J. Bauer, J. Masiero, R.S. McMillan, R.M. Cutri, R. Walker, E. Wright, P. Eisenhardt, D.J. Tholen, T. Spahr, R. Jedicke, L. Denneau, E. DeBaun, D. Elsbury, T. Gautier, S. Gomillion, E. Hand, W. Mo, J. Watkins, A. Wilkins, G.L. Bryngelson, A. Del Pino Molina, S. Desai, M.G. Camus, S.L. Hidalgo, I. Konstantopoulos, J.A. Larsen, C. Maleszewski, M.A. Malkan, J.-C. Mauduit, B.L. Mullan, E.W. Olszewski, J. Pforr, A. Saro, J.V. Scotti, L.H. Wasserman, et al., Neowise observations of near-Earth objects: preliminary results, Astrophys. J. 743 (2) (2011) 156. https://doi.org/10.1088/0004-637X/743/2/156

[7] T. Grav, A.K. Mainzer, J.R. Masiero, D.W. Dahlen, T. Spahr, W.F. Bottke, F.J. Masci, et al., The NEO surveyor near-Earth asteroid known object model, Planet. Sci. J. 4 (12) (2023) 228. https://doi.org/10.3847/PSJ/ad072e

[8] B.E. Clark, M.A. Barucci, X.-D. Zou, M. Fulchignoni, A. Rivkin, C. Raymond, M. Yoshikawa, L.T. Elkins-Tanton, H. Levison, et al., A brief history of spacecraft missions to asteroids and protoplanets, in: Primitive Meteorites and Asteroids, Elsevier, 2018, pp. 1–57. https://doi.org/10.1016/B978-0-12-813325-5.00001-X

[9] D.S. Lauretta, H.C. Connolly, J.E. Aebersold, C.M.O. Alexander, R. Ballouz, J.J. Barnes, H.C. Bates, C.A. Bennett, L. Blanche, E.H. Blumenfeld, S.J. Clemett, G.D. Cody, D.N. DellaGiustina, J.P. Dworkin, S.A. Eckley, D.I. Foustoukos, I.A. Franchi, D.P. Glavin, R.C. Greenwood, P. Haenecour, V.E. Hamilton, D.H. Hill, T. Hiroi, K. Ishimaru, F. Jourdan, H.H. Kaplan, L.P. Keller, A.J. King, P. Koefoed, M.K. Kontogiannis, L. Le, R.J. Macke, T.J. McCoy, R.E. Milliken, J. Najorka, A.N. Nguyen, M. Pajola, A.T. Polit, K. Righter, H.L. Roper, S.S. Russell, A.J. Ryan, S.A. Sandford, P.F. Schofield, C.D. Schultz, L.B. Seifert, S. Tachibana, K.L. Thomas-Keprta, M.S. Thompson, V. Tu, F. Tusberti, K. Wang, T.J. Zega, C.W.V. Wolner, the OSIRIS-REx Sample Analysis Team, et al., Asteroid (101955) Bennu in the laboratory: properties of the sample collected by OSIRIS-REx, Meteorit. Planet. Sci. 59 (9) (2024) 2453–2486. https://doi.org/10.1111/maps.14227

[10] D.W. Mittlefehldt, Asteroid (4) vesta: I. The howardite-eucrite-diogenite (HED) clan of meteorites, Geochemistry 75 (2) (2015) 155–183. https://doi.org/10.1016/j.chemer.2014.08.002

[11] R.L. Forward, Statite—A spacecraft that does not orbit, J. Spacecr. Rockets 28 (5) (1991) 606–611. https://doi.org/10.2514/3.26287

[12] D.A. Spencer, L. Johnson, A.C. Long, Solar sailing technology challenges, Aerosp. Sci. Technol. 93 (2019) 105276.

[13] Y. Tsuda, O. Mori, R. Funase, H. Sawada, T. Yamamoto, T. Saiki, T. Endo, K. Yonekura, H. Hoshino, J. Kawaguchi, et al., Achievement of IKAROS—Japanese deep space solar sail demonstration mission, Acta Astronaut. 82 (2) (2013) 183–188. https://doi.org/10.1016/j.actaastro.2012.03.032

[14] P. ZHAO, W.U. Chenchen, L.I. Yangmin, Design and application of solar sailing: a review on key technologies, Chin. J. Aeronaut. 36 (5) (2023) 125–144. https://doi.org/https://doi.org/10.1016/j.cja.2022.11.002

[15] J.R. Mansell, J.M. Bellardo, B. Betts, B. Plante, D.A. Spencer, et al., LightSail 2 solar sail control and orbit evolution, Aerospace 10 (7) (2023) 579. https://doi.org/10.3390/aerospace10070579

[16] A.N. Alvara, L. Lee, E. Sin, N. Lambert, A.J. Westphal, K.S.J. Pister, BLISS: interplanetary exploration with swarms of low-cost spacecraft, Acta Astronaut. 215 (2024) 348–361.

[17] X. Zeng, J. Li, H. Baoyin, S. Gong, Trajectory optimization and applications using high performance solar sails, J. Spacecr. Rockets 48 (3) (2011) 491–505.

[18] G. Mengali, A.A. Quarta, Solar sail trajectories with piecewise-constant steering laws, Aerosp. Sci. Technol. 13 (8) (2009) 431–441. https://doi.org/10.1016/j.ast.2009.06.005

[19] B. Dachwald, Optimal solar-sail trajectories for missions to the outer solar system, J. Guid., Control, Dyn. 28 (6) (2005) 1187–1193. https://doi.org/10.2514/1.13308

[20] I. Carnelli, B. Dachwald, M. Vasile, Evolutionary neurocontrol: a novel method for low-thrust gravity-assist trajectory optimization, J. Guid., Control, Dyn. 32 (2) (2009) 615–624. https://doi.org/10.2514/1.32633

[21] L. Johnson, E. Betts, A. Heaton, C. Jones, L. McNutt, M. Pruitt, J. Stott, D. Wallace, R. Wilson, J. Castillo-Rogez, G. Lantoine, C. Seybold, T. Sweetser, et al., Near Earth asteroid scout - mission update, 36th Annual Small Satellite Conference (2022).

[22] M. Vergaaij, C.R. McInnes, M. Ceriotti, Economic assessment of high-thrust and solar-sail propulsion for near-earth asteroid mining, Adv. Space Res. 67 (9) (2021) 3045–3058.

[23] J.H. Holland, Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, Mich., 1975.

[24] J.H. Holland, J.H. Miller, Artificial adaptive agents in economic theory, Am. Econ. Rev. 81 (1991) 365–371. https://api.semanticscholar.org/CorpusID:17734589.

[25] D.E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, 1989.

[26] L. Davis, Handbook of Genetic Glgorithms, Thompson Computer Press, 1991.

[27] C. Onwubiko, Introduction to Engineering Design Optimization, Prentice Hall, 2000.

[28] D.E. Goldberg, K. Deb, Special issue on genetic algorithms, Comput. Methods Appl. Mech. Eng. 186 (2–4) (2000) 121–124.

[29] NASA, What is the Deep Space Network?, 2023, (NASA Space Communications and Navigation Program). Accessed: [Insert date], https://www.nasa.gov/directorates/somd/space-communications-navigation-program/what-is-the-deep-space-network/.

[30] NASA's Voyager 1 Resumes Sending Engineering Updates to Earth — jpl.nasa.gov, (https://www.jpl.nasa.gov/news/nasas-voyager-1-resumes-sending-engineering-updates-to-earth/). [Accessed 02-12-2024].

[31] C.R. McInnes, Solar Sailing: Technology, Dynamics and Mission Applications, Springer Science & Business Media, 2004.

[32] L. Rios-Reyes, D.J. Scheeres, Generalized model for solar sails, J. Spacecr. Rockets 42 (1) (2005) 182–185. https://doi.org/10.2514/1.9054

[33] T.I. Zohdi, Mechanistic modeling of swarms, Comput. Methods Appl. Mech. Eng. 198 (21–26) (2009) 2039–2051. https://doi.org/10.1016/j.cma.2008.12.029

[34] T.I. Zohdi, Multiple UAVs for mapping: a review of basic modeling, simulation, and applications, Annu. Rev. Environ. Resour. 43 (1) (2018) 523–543. https://doi.org/10.1146/annurev-environ-102017-025912

[35] T.I. Zohdi, The game of drones: rapid agent-based machine-learning models for multi-UAV path planning, Comput. Mech. 65 (1) (2020) 217–228. https://doi.org/10.1007/s00466-019-01761-9

[36] T.I. Zohdi, A digital-twin and machine-learning framework for ventilation system optimization for capturing infectious disease respiratory emissions, Arch. Comput. Methods Eng. 28 (6) (2021) 4317–4329. https://doi.org/10.1007/s11831-021-09609-3

[37] G.C. Calafiore, L. El Ghaoui, Optimization Models, Cambridge University Press, 2014.

[38] D. Bertsekas, Nonlinear Programming, second ed., Athena Scientific, Belmont, Massachusetts, 1999.

[39] D. Luenberger, Introduction to Linear & Nonlinear Programming, Addison-Wesley, Menlo Park, 1974.

[40] B. Fu, E. Sperber, F. Eke, et al., Solar sail technology—A state of the art review, Prog. Aerosp. Sci. 86 (2016) 1–19. https://doi.org/10.1016/j.paerosci.2016.07.001